LEVEL II

12

CMU-CS-81-125

Comparative Study of Nonlinear Time Warping Techniques in Isolated Word Speech Recognition Systems

A. Waibel, B. Yegnanarayana

17 June 1981

Carnegie-Mellon University
Computer Science Department

DEPARTMENT
of
COMPUTER SCIENCE

Carnegie-Mellon University

CMU-CS-81-125

Comparative Study of Nonlinear Time Warping Techniques in Isolated Word Speech Recognition Systems

A. Waibel, B. Yegnanarayana

17 June 1981

Carnegie-Mellon University
Computer Science Department

F33615-78-C-1551.
ARPA Order-3597

This research was sponsored in part by the National Science Foundation, Grant MCS-7825824 and in part by the Defense Advanced Research Projects Agency (DOD), ARPA Order No. 3597, monitored by the Air Force Avionics Laboratory Under Contract F33615-78-C-1551.

Table of Contents

Abstract

In this paper we present the description of an isolated word recognition system and a discussion of various design choices that affect its performance. In particular, we report experimental results aimed at evaluating several methods to optimize the performance of dynamic warping algorithms. Three major aspects that have been suggested in the literature have been investigated: (1) relaxation of the boundary conditions to allow for inaccurate begin-end time detection, (2) choice of warping algorithm, e.g., Itakura asymmetric, Sakoe and Chiba symmetric, Sakoe and Chiba asymmetric, and (3) choice of an appropriate warping window to restrict computation to a minimum needed for best recognition results. Recognition results were tested on two vocabularies: the digits and a highly confusable subset of the alphabet (e.g., e, b, d, p, t, g, v, c, z). (4) The relaxation of the boundary conditions degraded the performance of the confusable subset and the digits. (5) The asymmetric Itakura algorithm yielded better results for the confusables, while we obtained slightly better results for the digits using the symmetric Sakoe and Chiba algorithm. (6) The choice of a 100-ms warping window appears to be optimal for both vocabularies used.

1. Introduction

Speech recognition is an important step towards more natural form of man-machine communication. In many administrative or industrial environments the use of machines, in particular computers, require prior knowledge and experience to operate these machines. Alternatively, situations exist in which particular modes of data entry (typing on a keyboard) are not available (e.g., telephone applications like directory assistance) or not feasible (e.g., if a human user needs his hands for other tasks and typing is impractical) or simply too slow (human speech transmits information at a significantly higher rate than typing). Most applications can be seen to point in the direction of bending the capabilities of machines to the needs of a human user rather than expecting a user to invest time, interest, knowledge and skills to make use of computers.

Speech is the most common form of human communication. It is desirable to provide this most natural form of communication in man-machine communication as well. Speech recognition thus plays an important role in making computers an integral part of every day life. For a variety of applications, speech recognition is already available, and increased capabilities are under development and can be expected to enter the public domain in the near future[1,2,3].

Although it has been shown that sophisticated speech understanding systems can yield a high degree of performance[45] and that efficient hardware implementations for such sytems can be developed, the need for better limited vocabulary speech recognition systems has become apparent. Such systems are both useful for a variety of practical applications and as a way to finding solutions to the problems of speech recognition at the signal level. The fact that human spectrogram readers can achieve a high degree of recognition accuracy even for nonsense utterances (i.e., in the absence of syntactic and semantic information)[6] is an indication that much improvement for any recognition system can still be expected to come from a better understanding of the recognition process at the signal level.

In the present study we are mainly concerned with issues connected with the development of an isolated word recognition system. Our hope is to extend the notions developed here to achieve further improvements, greater computational efficiency, speaker independent operation and the capability for connected speech input in the near future.

Fig.1 depicts an overview of the main functional parts of the system. The main purpose of the "Front End" is to digitize and parametrize the incoming speech data to provide a compressed representation of the speech signal that minimizes the storage allocation and the computational efforts needed in subsequent modules, thus eliminating irrelevant or redundant information, while preserving all relevant information. The module labeled "Matching" serves the purpose to extract and appropriately weight discriminatory cues in the process

of matching the incoming unknown test token with a reference token provided in the reference template data base. Since each of the modules still holds great potential for further improvements, all modules are loaded under a flexible research oriented supervisor, "Cicada". Cicada allows for the integration of experimental ideas, extensions of the recognition system, and for great ease of creating test environments for experimental runs of varying scope in a very convenient way. It thus provides both the generality and flexibility that is desirable for a research system, as well as reducing the implementational efforts needed to evaluate alternate recognition methods. More detailed information about Cicada can be found in[7]. In the following we limit our discussion to the design of the front end and to the design and optimization of the recognition algorithm ("Matching").

In the following sections several signal processing issues relevant to speech recognition will be discussed followed by a description of the design of the Front End, including a novel approach for automatic begin end detection. Subsequently, a detailed presentation of various recognition algorithms suggested in the literature or developed in the process of our investigations will be given. These algorithms were tested in three experiments that were run exhaustively over our entire data base. Optimization results and conclusions from this study will be found in the last chapters.

2. Signal Processing for Speech Recognition

2.1 The Choice of Speech Signal Representation

The main problem in speech recognition is the identification of common characteristics among several utterances of the same unit (word or sentence). Speech recognition by humans takes place by detecting certain key features in an utterance. It is therefore necessary to determine these features, called auditory hints. Although speaker and context dependent, there are several auditory hints which can be extracted by signal processing and can be utilized effectively in a recognition process.

Spectral representation of speech is the most widely used method in speech recognition. Other features such as energy, zerocrossings, pitch and duration are also used to supplement the spectral information. Evidence of the importance of spectral information in preserving speech information has been provided by several successful analysis-synthesis systems and also by spectrograms. Speech produced by linear prediction (LPC) or filter bank representation of spectral energy is highly intelligible, although it suffers from lack of naturalness. This lack of naturalness is due to poor representation of source characteristics in the synthesis part. Careful training of spectrogram reading enables one to identify most speech features needed for recognizing an utterance. Since in a speech recognition system the objective is to recognize only but not to reproduce, it seems that the gross spectral information is adequate for this purpose.

Although spectral representation forms the basis for both speech bandwidth compression systems as well as speech recognition systems, the requirements of the representation vary widely in both cases. In a speech bandwidth communication system the signal should be represented so as to reproduce as many temporal details as possible. The objective in this case is to produce a synthetic signal which resembles the originally very closely in perceptual quality. In other words, all the variability of speech and speaker will have to be preserved as far as possible. The processing therefore aims at representing all this information in a small number of parameters. The table below summarizes the differences in the requirements of signal processing for speech communication and recognition. The problem of signal processing for speech recognition, therefore, consists of reducing the variance while preserving the auditory hints. The auditory mechanism has the remarkable ability to detect sharp changes in the signal and ignore even long durations of significant energy regions, based on context. The concept of these auditory hints is probably responsible for human speech recognition across several utterances and speakers without prior training of a particular individual speaker.

Spectrogram reading experiments suggest several interesting clues for design of speech recognition systems. The results of the experiments demonstrates that the acoustic signal contains a great deal of phonetic

Table 2-1: Requirements of Speech Processing:

Communication	Recognition
1. Necessary to reconstruct the signal waveform.	1. Not necessary.
2. Speaker variability to be preserved.	2. Not necessary.
3. Speech variability to be preserved.	3. Variability to be suppressed.
4. Perceptual characteristics of speech and speaker information are needed (source characteristics).	4. Need to preserve perceptual characteristics of speech information only.
5. Usually vocal tract model based analysis . (production based).	5. Auditory hints based (perception based).
6. Representation problem.	6. Pattern matching problem.
7. Each utterance is dealt with independently.	7. Features common to multiple repetitions of a word are needed.

information which can be captured by rules. The first thing to realize is that spectrogram displays only gross spectral features and the suprasegmental features like intensity duration and pitch. All the available information is used both globally and locally to recognize an utterance. The spectral information is compressed to a low dynamic range of about 15-20 dB in a spectrogram. Despite the crude nature of displayed information the high recognition performance is a result of the reader's ability to use only the relevant information at each level (global and local). In particular, many times even the high energy spectral information is not considered, as for example, the energy below about 400 Hz.

It is also interesting to note that very little speaker dependent information is captured by a spectrogram reader. That means only features that are mainly speaker independent are used for recognition. The reader's ability to recognize speech patterns even in the presence of some multiplicative or additive spectral distortions suggest that the key temporal and spectral features are small and robust and probably context-dependent. A spectrogram-like representation of the speech signal would thus appear to be adequate.

The above discussion also suggests that a uniform vocal-tract modeling approach like linear prediction analysis and matching using linear prediction coefficients may not be very suitable for a practical speech recognition system. In a spectral representation of LPC type the features corresponding to high energy level

are emphasized over the entire frequency range. In a uniform matching technique like LPC metric all the spectral information is used for determining the class of a given segment of speech. In other words selective frequency domain matching for different classes of sounds, as used intuitively in a spectrogram reading, is not possible. Moreover, distortions alter the LPC type of representation in a signal dependent manner. Uniform processing over time domain, like fixed frame rate analysis, also prevents selective processing depending on context. The uniform representation of spectral information is also highly speaker dependent and this dependence cannot easily be altered by simple transformations.

Comparing the three modes of spectral representations, namely, uniform modeling approach as in LPC, spectral values as in short-time spectral analysis and filter bank output reveals several distinct characteristics in each mode. The characteristic of LPC spectrum is that it approximates the peaks in short-time spectrum better than valleys and this provides an efficient representation of the spectral envelope. It is thus ideally suited to information storage and speech synthesis. Short-time spectral values give a detailed description of the spectrum for purposes of analysis of vocal-tract transfer function and its excitation. Filterbank output contains the temporal variation of signal energy in selected frequency bands, thus it provides a description of the averaged characteristics in each band.

From recognition point of view, selective processing in time and frequency domains holds the key to success, as evidenced from the spectrogram reading experiments. A system should recognize an unknown utterance, may be spoken by a different speaker, under different conditions of environment, and at different times. Thus the statistical properties of the factors causing variability are not available, and even if available, they are not useful. The features for recognition, therefore, should be robust under various conditions of speech production.

Recent results[8] indicate that the choice of mel-frequency cepstral coefficients yields better recognition performance over linear frequency cepstral coefficients , LPC and reflection coefficients. The success of the mel-frequency cepstral coefficients is most likely due to its virtue of modeling the perceptual behavior of the auditory system more closely, by simulating the variations with frequency of the critical bands on the basilear membrane. An additional advantage of using cepstral coefficients as evidenced in our own informal experimentation and from the results by Davis and Mermelstein[8] is that the use of only 6 coefficients seems to suffice to represent all relevant information. In informal experimentation we have used two parametric representations: 16 coefficients derived from bandpass-filtering the signal according to the mel-frequency scale (see table below) and 6 cepstral coefficients derived from this filterbank output. Informal observation did not reveal significant differences between the two representations. The advantages of using filterbank coefficients are that frequency selective recognition schemes can be easily implemented, the effects of filterbank coefficients on recognition can readily be conceptualized and that hardware filterbank

implementations are currently realized in many commercially available systems. For the present comparative study mel-frequency filterbank coefficients have been chosen for the spectral representation. A detailed outline of the signal processing performed in the front end of the recognition system is given below.

2.2 Description of the Scheme

In Fig.2 the functional blocks of the isolated recognition system is depicted. Speech data played back from a cassette tape recorder was low pass filtered to 4500 Hz and sampled at 10 KHz rate. The samples were stored as 16 bit numbers. A preliminary word boundary detection based on amplitude was used to determine the signal region.

A frame size of 20 msec is chosen for analysis. The data in the analysis interval are multiplied with a Hamming Window. The discrete Fourier transformation (DFT) of the windowed data is computed using a 256 point FFT. The 56 additional points are set to zero. The spectrum is computed by summing the squares of real and imaginary parts of the DFT. In the resulting spectrum the sample numbers from 1 to 128 define the frequency range 0-5 kHz.

The spectral values on the 0-5 kHz range are reduced to 16 values by using an approximate mel frequency scale. Table 2-2 gives the mel frequency sample index and the corresponding frequency intervals over which the spectral values are added to obtain the mel scale spectral value. Only half of the common spectral value between adjacent intervals is considered.

After subtracting the background noise the spectral values on the melscale are represented as integer number on dB scale i.e. The log mel spectral values in dB are give by:

$$L_i = 10 \log_{10} m_i \quad i = 1,.....16$$

For matching, a frame rate of 100 frames per second is chosen. To further compress the data and to normalize for overall energy level of the signal, 15 coefficients are computed by differencing the adjacent spectral values across frequency.

Two frames of two different utterances (namely the unknown and the reference are compared by computing the squared Euclidean distance between the 15 filterbank coefficients of the two utterances to be matched, i.e.,

$$d_{ij} = \Sigma_{k=0}^{15} [M_i(k) - M_j(k)]^2$$

where $\{M_i(k)\}$ and $\{M_j(k)\}$ are the mel cepstral coefficients for ith and jth frames respectively.

Table 2-2: Reduction of Spectral Data to Mel Frequency scale:

[s(i) is the spectral value at the ith sample]

[m(i) is the ith spectral coefficient on the mel scale]

Index on the mel frequ. scale i	Spectral Coefficients on mel scale	Frequency Interval
1	m(1) = s(1)+s(2)+s(3)/2	0-117 Hz
2	m(2) = s(3)/2+s(4)+...+s(6)+s(7)/2	117-273 Hz
3	m(3) = s(7)/2+s(8)+...+s(10)+s(11)/2	273-429 Hz
4	m(4) = s(11)/2+s(12)+...+s(14)+s(15)/2	429-585 Hz
5	m(5) = s(15)/2+s(16)+...+s(18)+s(19)/2	585-742 Hz
6	m(6) = s(19)/2+s(20)+...+s(22)+s(23)/2	742-898 Hz
7	m(7) = s(23)/2+s(24)+...+s(26)+s(27)/2	898-1054 Hz
8	m(8) = s(27)/2+s(28)+...+s(30)+s(31)/2	1054-1210 Hz
9	m(9) = s(31)/2+s(32)+...+s(35)+s(36)/2	1210-1406 Hz
10	m(10) = s(36)/2+s(37)+...+s(41)+s(42)/2	1406-1640 Hz
11	m(11) = s(42)/2+s(43)+...+s(48)+s(49)/2	1640-1913 Hz
12	m(12) = s(49)/2+s(50)+...+s(57)+s(58)/2	1913-2265 Hz
13	m(13) = s(58)/2+s(59)+...+s(68)+s(69)/2	2265-2695 Hz
14	m(14) = s(69)/2+s(70)+...+s(81)+s(82)/2	2695-3202 Hz
15	m(15) = s(82)/2+s(83)+...+s(97)+s(98)/2	3202-3827 Hz
16	m(16) = s(98)/2+s(99)+...+s(116)+s(117)/2	3827-4570 Hz

2.3 Begin-End Frames Detection

For matching two isolated utterances or words, the end-points of the utterance must be known accurately. It is important that the automatic detection of the endpoints is performed accurately, since, as we shall see, confusion in the subsequent recognition is the immediate consequence and possible recovery from misrecognized endpoints is difficult. The difficulties in automatic endpoint detection arise from the attempt to discriminate between speech (which inludes weak frication noises as in the word "FIVE") and non-speech

signals, such as background noise, speaker or system generated clicks and pops. In addition, the algorithm has to decide whether two intervals of speech signal belong together (as in "SIX" and "X", where the fricative part of the final [s] is separated from the rest of the utterance by the stop closure). These features of the incoming signal had to be taken into consideration in the design of the algorithm. Several methods were proposed for end-point detection of an utterance, but all of them use time-domain parameters such as amplitude, energy, zero-crossing, etc. Since most systems use spectral features for recognition, it would be useful to have an end-point detection algorithm based on spectral values. Some of the advantages in using spectral parameters over time-domain parameters are:

1. They are less sensitive to noise.
2. It is easier to fix thresholds.
3. The decisions can be made independent of absolute amplitude levels of the signal.
4. Since the spectral values are obtained by reducing the data to mel scale, the decisions will be robust.

2.3.1 Parameters:

The following parameters are used for end-point detection:

1. Average level in dB (L).
2. Difference between high frequency and low frequency levels in dB (L_d).
3. Background noise level in dB (L_0).

For computing values of L and L_d, the first and the sixteenth log spectral values on the mel scale are ignored. This is because the first value is strongly dependent on breath noise and the last (sixteenth) value is very susceptible to additive noise. The background noise level is computed as follows:

1. Select the lowest 5 of the first 10 frames by arranging them in increasing order of their average overall level. This will take care of impulsive noise like clipping.
2. Determine the average of L and denote it by L_1.
3. Repeat steps(1) and (2) for the last 10 frames and denote the resulting average value as L_2.
4. Choose the lower of the levels L_1 and L_2 as the background noise level L_0.
5. Compute the average of L_d over the five frames used to compute L_0 and denote it by L_{od}.
6. If L_1 and L_2 are higher or lower than some "reasonable" background noise levels, a value of 55 dB is assumed for L_0. This situation may arise if the signal begins and ends outside the boundaries of the data file.

2.3.2 Notation and default values for thresholds:

1. nf = total number of frames
2. b1 = L
3. b2 = L_d
4. b1t = L_o
5. b2t = L_{od}
6. lowt = 8 dB (lower threshold on level)
7. hght = 18 dB (higher threshold on level)
8. zct - 10 dB (threshold on hf-lf level)
9. nft - 0 (threshold for number of frames for smoothing decisions).
10. incr 1 = 5 dB (increment in level threshold after 30 frames).
11. incr 2 = 5 dB (decrement in hf-lf threshold after 30 frames).
12. hghtx = 15 dB (threshold to determine genuine speech interval).
13. decision (d) is -1 for silence and +1 for signal and 0 for intermediate cases.

2.3.3 Decisions:

Initialize the first 5 and last 5 frames decisions to silence, i.e., -1. Starting from nft up to nf-nft use the following logic to determine the silence/signal frames.

1. If b1>(b1t+hght), then d = 1.
2. If b1<(b1t=lowt) and b2<(b2t+zct), then d = -1.
3. If b1<(b1t+lowt) and b2≥(b2t+zct), then d = 0.
4. If b1 lies in the range b1t+lowt and b1t+hght and b2<(b2t+act), then d = 0.
5. If b1 lies in the range given in (4) and b2≥(b2t+zct), then d = 1.

2.3.4 Smoothing the decisions:

The above decisions are smoothed using an 11 frame window. If the sum of decisions in the window is less than or equal to 0, then the decision is set to -1. Otherwise, the decision is set to +1.

In general there can be more than one interval like in utterances /8/ and /h/. To check the genuineness of the additional intervals, their average level is compared with a threshold (blt+hghtx in this case). If the level exceeds the threshold, then the end of the utterance is the end of the second interval. Otherwise, the end of the utterance is the end of the first interval itself. Extensive testing and comparison with manually set endpoints was performed to choose the best thresholds.

3. Matching Methods for Isolated Word Recognition

3.1 Introduction

Although research in speech recognition has advanced in recent years to a state in which speaker independent connected speech recognition has become feasible, several questions relating to design choices of isolated word recognition systems have remained unanswered. These design choices affect both recognition accuracies and computational efficiency drastically and it is important to carefully investigate these issues before deciding in favor of any such designs. Much attention has already been devoted to the optimal choice of parametric representation of the spectral information and to the choice of the algorithm used to perform time alignment between an unknown test utterance and a given reference template. Several techniques also have been suggested to improve recognition accuracy in the presence of errors in the begin-end time detection of the utterance. Preliminary experimentation with an isolated word recognition system has led us to define--in agreement with many previous studies --several constraints or problem areas causing severe differences in recognition accuracies:

1. the vocabulary being used
2. speaker variations (cooperative, non-cooperative speakers)
3. begin-end time detection
4. reference template selection

Although these problem areas may seem obvious, most experimental studies have investigated speech recognition techniques keeping the above variables fixed, i.e., one vocabulary, selected speakers, manual or semi-automatic begin-end time determination. In the present study we will attempt to account for these variables and attempt to select optimal design choices. In three experiments we are particularly concerned with the choice of dynamic programming algorithm, methods to relax boundary conditions to deal effectively with incorrect begin-end detection, and the optimal choice of a dynamic programming search space to increase computational efficiency.

3.2 Nonlinear time alignment by dynamic programming

Many studies have already investigated[9] the problem of how to most effectively align an incoming unknown test-token to a known reference token or reference template. The goal in applying any such time alignment procedure is to optimally account for durational variations of two different utterances of the same

word. The fundamental problem in the design of such a matching scheme is to implicitly tolerate variations between two tokens that bear no phonetic relevance and to penalize when variations are present that are of importance in discriminating between utterances in a linguistically meaningful way. Nonlinear time warping by dynamic programming has been shown to incorporate these goals to some degree in a very elegant way. It's superiority over linear time warping methods[10,11] is due to the fact that it allows for an unevenly distributed (nonlinear) "stretching" and "compressing" along the time axes of the utterances to be matched. This way it can account for the nonlinear changes in duration of the various phonetic subunits of syllables or words. The elegance of dynamic programming is that we obtain this nonlinear treatment without the necessity of segmentation, and thus avoid this additional source of errors.[1]

The basic principle of dynamic programming can be considered to be a mapping of the time-axis of a speech pattern A onto the time-axis of a pattern B in such a way that the resulting dissimilarity is minimized. Adopting the notation of Sakoe & Chiba[10], this can be formalized as follows

Let us assume the speech patterns A and B to be two sequences of parameter vectors describing the signal properties of the utterances at a given instant (frame) in time, then we can write

$$A = a_1, a_2 \ldots\ldots a_i \ldots a_I \quad \text{and}$$

$$B = b_1, b_2 \ldots\ldots b_j \ldots b_J$$

We will furthermore illustrate the mapping procedure as a search space in an i-j plane, where the horizontal axis i represents the time axis of the test token and the vertical axis j represents the time axis of the reference token (see Fig.3). For each point P(i,j) in this warping plane, we define a distance or dissimilarity measure d(i,j). The goal of nonlinear time warping is to find the path (with path index k) through this plane whose cumulative distance

$$D(A,B) = \Sigma_{k=1}^{K} d(i(k),j(k)) \qquad (3.1)$$

is minimal. At the endpoint P(I,J), this cumulative distance will then be considered as the dissimilarity score for the match between utterances A and B and will subsequently serve as a decision criterion for the recognition.

Introducing a path weighting function w(k),(3.1) can be rewritten as

---

[1] It should be noted that endpoint detection can be considered a still remaining segmentation problem. As we shall see, it heavily affects the outcome of the recognition.

$$D(A,B) = \text{MIN}\,[\Sigma_{k=1}^{K} d(i(k),j(k))\cdot w(k)/\Sigma_{k=1}^{K} w(k)] \qquad (3.2)$$

where f symbolizes all possible paths through the warping plane. The expression in the denominator serves to normalize the dissimilarity score to render it independent of the number of points on the search path k. For the case of $w(k)=1$, for example, $\Sigma_{k=1}^{K} w(k)$ simply reduces to K; and D(A,B) is simply the average distance, averaged over the entire search path.

For the practical application of speech pattern matching, the goal of providing great flexibility of the search path (to obtain a minimal dissimilarity score) and the desire to only allow linguistically meaningful compressions and expansions of the speech signal have to be traded off by imposing constraints or restrictions on the search path. In this study we are interested in comparing the performances of the asymmetric warping algorithm proposed by Itakura[12] and the symmetric and asymmetric cases of the (P=1) warping algorithms by Sakoe & Chiba.

For the purpose of this comparative study we redefine the constraints somewhat differently for the reported versions. These alterations have become necessary to keep the variables fixed across the various conditions investigated. These alterations will not affect the validity of the conclusions we are seeking. For all the three algorithms the following constraints have been applied:

3.2.1 Monotonicity:

$$i(k\text{-}1) \le i(k) \qquad (3.3)$$

$$j(k\text{-}1) \le j(k)$$

3.2.2 Boundary conditions:

$$i(1)=1,\ j(1)=1 \qquad (3.4)$$

$$i(K)=I,\ j(K)=J$$

(In the next section, we will investigate methods of relaxing this condition.)

3.2.3 Adjustment window or slope constraint:

All algorithms under consideration in this experiment implicitely define an identical slope constraint. This means that, the search path in Fig. 3 is restricted to stay within the limits given by slope 2 and slope 1/2. This restriction keeps the expanding and compressing function of the warp within linguistically meaningful limits. Thus horizontal or vertical paths that imply skipping several frames in one of the utterances will not be possible and the presence of different segments in the test or the reference token will result in a forced higher total dissimilarity score and hence be a good indication of a poor match. These slope constraints (1/2 and 2) together with the boundary conditions (3.4) restrict the search path to stay within a parallelogram illustrated in Fig. 3. In some recognition schemes, the definition of an adjustment window that defines a meaningful search space has been necessary, particularly, when the above mentioned slope constraints were lacking[10]. Alternatively, the use of a window can prove useful since it eliminates redundant computation. This issue will be discussed later in this paper. For the first experiment, the slope constraint will serve the purpose of defining the search space as shown in Fig. 3. It is consequence of the continuity conditions and the warping functions as described below.

3.2.4 Continuity conditions and warping functions:

We have already noted before that our total cumulative distance D(A,B) is the sum of the distances between time frames of the test and of the reference utterance along the "best" path through the warping plane. It remains to define an algorithm that will choose the best path, namely a path that will result in a low value of the total distance D(A,B) if A and B are the same utterances. For each point in the search space the cumulative distance along the least expensive path up to this point is computed. More formally, this can be expressed in the dynamic programming (DP) equation:

$$g_k(i(k),j(k)) = \min_{i(k-1),j(k-1)} \left[g_{k-1}(i(k-1),j(k-1)) + d(i(k),j(k)) \cdot w(k) \right] \qquad (3.5)$$

For the three algorithms this can be accomplished in the following manner (refer to Fig. 4):

1. *Warp 1 (Itakura, asymmetric)*:

The continuity condition

$$\begin{aligned} j(k) - j(k-1) &= 0,1,2 \qquad (j(k-1) \neq j(k-2)) \\ &= 1,2 \qquad (j(k-1) = j(k-2)) \end{aligned} \qquad (3.6)$$

implies the upper and lower bounds of the slope constraints, namely the values 1/2 and 2. The DP-equation for this algorithm can be written as

$$g(i,j)= \min\left\{\begin{matrix} g(i-1,j-1) \\ g(i-1,j-2) \\ \min\limits_{g(i-2,j-2)} g(i-2,j-1) + d(i-1,j) \end{matrix}\right\} + d(i,j) \tag{3.7}$$

Notice that the weighting function w(k) in this case is 1.

It is also insightful to note that this algorithm allows for frames in the reference template to be skipped entirely if g(i-1,j-2) in the DP-equation happens to be minimal. Thus time alignment is achieved by compressing and expanding the time axis of the reference token.

2. *Warp 2 (Sakoe & Chiba symmetric)* Here the somewhat different continuity conditions

$$i(k)-i(k-1) \leq 1 \text{ and } j(k)-j(k-1) \leq 1 \tag{3.8}$$

combined with the DP equation in this case

$$g(i,j)= \min\left\{\begin{matrix} g(i-1,j-2) + 2d(i,j-1) + d(i,j) \\ g(i-1,j-1) + 2d(i,j) \\ g(i-2,j-1) + 2d(i-1,j) + d(i,j) \end{matrix}\right\} \tag{3.9}$$

yields again the same slope constraints and thus limits the warp to the same search space as warp 1. Here the weighting function w(k) is given by

$$w(k)=(i(k)-i(k-1))+(j(k)-j(k-1)) \tag{3.10}$$

This weighting was chosen for this symmetric algorithm to make two paths between points A and B equally likely. This would not be the case for w(k)=1, since in this case, the diagonal path would always be favored (Fig.5), because of its smaller number of distances. By this method no frames are skipped and time alignment is obtained by appropriate time axis compression of the reference or the test token only.

3. *Warp 3 (Sakoe & Chiba asymmetric)* The continuity condition for this algorithm is identical to the one of Warp 2. The DP-equation is given by:

$$g(i,j)= \min\left\{\begin{matrix} g(i-1,j-2) + (d(i,j-1) + d(i,j))/2 \\ g(i-1,j-1) + d(i,j) \\ g(i-2,j-1) + d(i-1,j) + d(i,j) \end{matrix}\right\} \tag{3.11}$$

Again we obtain the same slope constraint.

The weighting function w(k) for the asymmetric warp in its original form is given by

$$w(k) = (i(k) - i(k-1))$$

Since w(k) in this case results to ∅ whenever i(k)=i(k-1), i.e., when a vertical path is attempted, the cumulative distance obtained from (3.5) would entirely disregard the distance associated with that point. DP-equation (3.11) is therefore a compromise that has been reported to yield better performance[10]. An equal share of the weight of 1/2 is simply given to each of the two distances involved. In this manner we obtain an algorithm that achieves time alignment (like Warp 2) by time axis compression only. In this case, however, (unlike Warp 1) compression of both the time axis of the reference and the test token can take place.

3.3 Relaxing the Boundary Constraints

It has been noted before that the presence of errors in the automatic begin-end time detection remains a source of drastic degradations in recognition performance. Although the development of speaker adaptive, background noise adaptive endpoint detection is in progress and might yield much improvement in this matter, it is desirable to perform the matching of the utterances in such a fashion that it is largely unaffected by minor inaccuracies in the endpoint detection. It can be seen from Fig. 3 that, for fixed boundary conditions, at the beginning and end of the match, i.e., in the extreme corners of the search space, little or no excursions of the search space are possible. This implies that in the presence of small deviations from the exact location of the endpoints, high distances will be computed at these points. The warping path thus will go through a few poor matches until proper alignment can be achieved. Particularly, in recognition tasks involving a vocabulary of high similarity, a small number of poorly matching time frames suffices to disturb the overall distance measure in such a way that recognition errors result.

Several methods have been proposed to account for these difficulties and we shall briefly introduce them. In all cases the primary goal is to allow some flexibility at the boundaries in order to avoid forcing poor matches. One possible method is to slightly deviate from the traditional concepts of dynamic programming and not use the endpoints of test and reference utterances as anchor points between which the time alignment has to take place, but rather to allow the search space to develop around the best matching path[13,14]. In this fashion the best match is continuously sought out of an unknown signal. Thus, it is not a match between two fixed length utterances but rather could be considered as moving a reference window through an unknown. This concept has been used to extend isolated word recognition schemes to word spotting applications[15] and to continuous speech recognition systems[16,15]. Recently Davis and Mermelstein[8] have also shown the usefulness of preliminary time alignment, in order to anchor the recognition on islands of reliability, namely prominent syllabic energy peaks, rather than on automatically or manually selected endpoints. This appears to be of particular importance when the test tokens are not read in isolation but are embedded in a phrase or sentence[8] and segmentation creates artificial boundaries.

In this paper we have investigated two alternate methods to account for endpoint inaccuracies. They both are conceptually aimed at relaxing the boundary constraints imposed by the warping algorithm. In the first

method, proposed by Rabiner et al.[14,13], this is achieved by allowing the start and end points to lie within a tolerance region δ on the vertical (reference) axis and 2δ on the horizontal (test) axis of the warping plain. Thus this modified warping algorithm spans the search space as shown in Fig. 6. Thus, given for example, an inaccurate starting point in the test or the reference utterance, the algorithm can skip up to δ or 2δ frames to align the test and the reference at the beginning and at the end. In spite of the superiority of this method over the constrained endpoint method in the case of the digit vocabulary, recent results by Rabiner on an alpha-digit vocabulary and preliminary results using a highly confusable subset of the alpha-digits show that under these conditions recognition rates actually deteriorate. The reason for this behavior is quite simple. By allowing several frames in the test and in the reference token to be skipped, the algorithm will conveniently skip over important short segments in cases where short important discriminatory acoustic information is contained right in the beginning or the end of an utterance. For the case of the alpha-digit vocabulary, for example, discrimination between "B" and "E" deteriorates by virtue of not constraining the algorithm to attempt to match the short formant transition region. Thus an overall low dissimilarity score might result and cause the utterance to be confused. While on one side the algorithm does yield better performance by lowering the dissimilarity score for "good" matches, it does not provide the second aspect, namely to penalize, i.e., increase, the dissimilarity score in the case of bad matches. An additional source of confusion here is due to the properties of the Itakura warping algorithm. Relaxing the boundary constraints on the test token (x-axis) will encourage a path that starts at the right-most allowable frame in the test-utterance, since at a given point P(i,j) and in the search space the path starting at this right most frame will be the summation over $i-\delta$ distances which usually is less than the summation over i distances on a path coming from the origin. To compensate we have informally attempted to use average distances instead of cumulative distances, but preliminary results have proven this idea to be unsuccessful. As an alternate design choice, therefore, a slightly modified method has been investigated. The relaxation of the boundary constraints has been restricted to the reference token. The new boundaries are thus (see Fig.6.c for illustration)

$$i(1) = 1,\ i(K) = I \text{ and}$$
$$1 \le j(1) \le \delta,\ J-\delta \le j(K) \le J$$

Here every frame in the test utterances will be matched in some way with the reference utterance and it is not possible to skip over information; yet, a certain tolerance in the choice of the starting point on the (y-axis) reference axis is given. This seems feasible in view of practical recognition systems, since the manual or semi-automatic choice of the endpoints of the reference utterance is a realistic possibility, while it is not for an incoming unknown test token.

This and the algorithm described above have been evaluated for δ of 3 and of 5 (i.e. 30 and 50 msecs, respectively).

3.4 Search Space Window

It has previously been stated that the warping algorithms used in this study span a search space in shape of a parallelogram by virtue of the slope constraints. It is reasonable to assume, however[13],[14], that the paths leading through the corners B and C in Fig. 7 are highly unlikely to occur in reality. Thus unnecessary computation is being performed at no gain and possibly loss of recognition accuracy. Computationally, the number of grid points in the search space is a good measure of costliness, since for each grid point the warp and the distance computation have to be performed. Reducing the search space as much as possible, therefore, is an efficiency constraint that has to be traded off or compared with the desire to achieve optimal recognition accuracy. Recognition can be expected to deteriorate if the search space is limited too severely. It has been noted that for some warping algorithms the definition of a search space delimiting adjustment window has become necessary. Such a window can be useful for the present algorithms also When superimposing a window onto the parallelogram of the warping search space, we obtain a new area, i.e., the parallelogram minus the corners (shaded regions in Fig. 7) at B and C. The amount of computational saving obtained by imposing this window constraint is dependent on the length of the two utterances to be matched. Clearly, if one utterance is significantly longer than the other, the parallelogram will become rather thin (in the limiting case I<J/2 or I>2J it will be non-existent and the warp can be aborted) and might lie within the preset window-width. To obtain useful estimates in this matter we have generated histograms of utterance lengths for different readings of a particular speaker and vocabulary. Fig. 13 through 15 show the histograms for the ten readings of the V1 vocabulary (see Table 1), the V2 vocabulary, and the alpha-digit vocabulary (all digits and the letters of the alphabet). From simple geometric considerations, the computational saving in % can easily be derived given the lengths of the test and the reference token and given the window widths. Together with the histograms, we can evaluate the average saving for a given window width and for a given speaker. Fig.16 shows the average saving for each speaker for the alpha-digit vocabulary, Fig.17 for V_1, and Fig.18 for V_2. For conceptual reasons we do not actually use the window width i(w) but rather the tolerance t (Fig. 7), a measure of the range of frames within which the match with the reference utterance is allowed to run ahead or lag behind the test utterance.

Notice that a tolerance of 0 implies linear time normalization or, in terms of Fig. 7, that only the grid points lying on the diagonal are computed and thus the saving is nearly 100%. In the other extreme, when the window width lies outside the warping parallelogram, no saving is obtained. The purpose of this experiment is to optimally trade off computational efficiency and recognition accuracy. More specifically, t was chosen to have the values 0, 3, 5, 8, and infinity, in other words, linear time normalization, a window of tolerance of ±30 msec, of ±50 msec, of ±80 msec, and no window at all.

4. Experimental Method

For our experimental investigation we have mostly assumed worst case conditions to test all of the above ideas for robustness and consistency.

4.1 Vocabulary

The principal vocabulary of interest for our recognition system are the alphadigits, i.e., the digits "one" through "zero" and the letters of the alphabet "A" through "Z". This vocabulary is not only very useful for a number of real life applications, but also provides us with a set of utterances, out of which subsets with varying degrees of discriminability can easily be defined. These subsets are of great interest since the acoustic similarities within such subsets point out the deficiencies of current speech recognition techniques. Here they serve to study the performance of the various techniques listed above separately, i.e., when the techniques are confronted with varying task domains. The particular two vocabularies (V_1 and V_2) that we used for this study are the ten English digits "ONE" through "ZERO" and the highly confusable subset of the alphabet, e.g., utterances that all end in the vowel [i] (see Table 1). Vocabulary V_2 is particularly interesting, since all relevant discriminatory information is contained in a short segment of less than 100 msec duration in the beginning of the utterance. The longer part of the utterance, the vowel part, on the other side, yields little or no additional information. In fact, without applying any segmentation or weighting function to a given matching procedure, the predominance of the vowel part, will increase confusability[17,18]. The vowel part in the utterance "B", for example, might match the vowel part in "P" better than what should be the correct choice, the reference template for "B". It is, therefore, reasonable to assume that the distribution of relevant discriminatory information over time is consistently different between the utterances of the vocabularies V_1 and V_2. Thus, rather than averaging over these differences, we consider these two vocabularies separately to increase the general validity of possible consistent results or to differentiate between them. Testing for robustness under the use of vocabularies of varying difficulty has recently been shown to be effective in finding generally applicable optimizations[19].

4.2 Speaker Variations

For the present study, no attempts of normalization over speaker variations are made. All eight speakers, four male (FA, MA, RP, JL) and four female (MS, DS, GG, SW), have been randomly selected. In our evaluation of the data obtained, we will therefore display these results for each speaker separately. As we shall see, quantitative as well as qualitative variations can be seen across speakers, thus rendering this separate treatment useful and insightful.

4.3 Endpoint Detection

As has been noted by many authors, the automatic determination of the endpoints of an utterance still remains a problem that consistently introduces a source of errors in any speech recognition system. Alternatively, many recognition errors can be eliminated by appropriately manually tuning the endpoints of an utterance. The human speech knowledge that implicitly is introduced by such manual tuning, however, renders comparisons between various recognition schemes difficult if not impossible, particularly if we rely on recognition rates as a measure of goodness of a specific method. We have therefore decided to perform our investigations under worst case conditions in this matter also, namely to use completely automatic endpoint-detection and thus allow for degraded recognition results due to errors in the endpoint detection. This procedure seems appropriate if we want to evaluate recognition schemes, such that conclusions might be robust enough to stand various real life applications. As a matter of fact, since we do not create or select reference templates independently, our recognition results will strongly reflect endpoint detection errors as we shall see. It should be noted here that, for the case of the utterance "eight", two different pronunciations are possible: one where aspiration noise follows the stop closure of the "t" and one simply ending with the stop closure, i.e., in which the closure is never released. These differences in the signal can be viewed as differences in pronunciation and, consequently, discrepancies in the automatically chosen endpoints cannot be classified as endpoint detection errors. A slight alteration that can be used to account for these discrepancies is to select two templates, one for each case. For the present study, however, we eliminated one of the pronunciations from consideration completely to simplify the experimental procedure.

4.4 Test and Reference Data

Each of the eight speakers read the entire alphadigit vocabulary a total of ten times ; two repetitions each day over a period of five days. The recordings were made in an office environment with a noise canceling microphone and a high quality tape recorder. We thus obtained a data base of 36 utterances X 10 sets (readings) X 8 speakers = 2880 test tokens to be used for our experiments.

The recorded data was passed through the front end of the recognition system as described previously. The input to the various algorithms investigated in this paper thus consisted of 15 spectral coefficients for every 10 msec speech and the automatically detected endpoints. Subsequently matching was performed as described below.

When running recognition experiments, it is clear that significant improvements can be achieved when appropriate reference templates are chosen. Rabiner et. al[1320] have shown that clustering techniques not only improve the reliability of speaker dependent recognition systems, but that they can be extended to be suitable

for speakers independent operation. Davis and Mermelstein[8] have recently proposed an iterative procedure that creates highly reliable reference templates. This can be achieved by averaging and time normalizing over a given set of training tokens. Li, Alleva, and Reddy[21] show that even a relatively simple selection mechanism suffices to pick out unambiguous and thus reliable reference tokens yielding a reduction in error rate by more than 1/2. The latter technique has the advantage of not incurring the danger of losing or deemphasizing acoustically and linguistically important information, such as air burst, glottal pulses, formant transitions, durational cues, and the like in the process of automatic averaging and normalizing. In the present study, however, we have decided to use each data set as reference once and match all the other nine sets against it. This method, employed by Sakoe and Chiba and others[1011] has the advantage of exhaustively utilizing all the data available and hence increasing the number of matches performed. It is hoped that in this fashion our results will be more robust and unaffected by separate problem areas such as speech variability. On the other hand, it should be noted that our results will reflect singular difficulties such as severe endpoint detection errors more readily, since each utterance misrepresented by its endpoints might now cause several mismatches to occur.

In summary, we have tested each condition by using eight speakers, two vocabularies, V_1 and V_2, and choosing each of ten data sets as reference. For each condition, speaker and vocabulary

10 (# of reference sets) X 9 (# of test sets matched with each reference set) X 10 or 9 (# of utterances in one test set for V_1 or V_2 respectively) = 900 recognitions (for V1), 810 recognitions (for V2)

were performed. Thus, each condition was tested by a total of:

8 (speakers) X [900 recognitions for V_1 + 810 recognitions for V_2] = 13680 recognitions.

5. Results and Discussion

5.1 Experiment I: Warping algorithms

The recognition results for the three warping algorithms and for all eight speakers are shown, for vocabulary V_1 in Table 2.a. and for vocabulary V_2 in Table 2.b. Significance testing by aligned ranks[22] was first performed to establish the significance of possible differences between the three algorithms. For the confusable vocabulary these differences were found to be significant at the $p < .025$ level, while for the digit vocabulary only $p < .3$ level significance was computed.

In addition, Wilcoxon paired comparison ranking was performed to establish the significance of the differences between the three warping algorithms. In good agreement with[10], superiority of warp 2 over warp 3 was established for both vocabularies, e.g. at the $p < .1$ level for V_1 and at the $p < .03$ level for V_2. As for warp 1, different results were found for V_1 and V_2. For V_1, warp 1 was seen to be superior to warp 2, ($p < .02$) while for V_2, warp 2 was found to be equivalent to or insignificantly ($p < .32$) better than warp 1. In order to understand this latter result, the strengths and weaknesses of both algorithms (warp 1 and warp 2) were investigated more carefully. In particular, let us focus on the differences between warp 1 and warp 2 as reflected by vocabulary V_2. Two typical confusion matrices (speaker: DS) for both algorithms are displayed in Table 3. All numbers off the diagonal are numbers of mismatches. The column labeled "Total" indicates the number of times a particular utterance was confused. Table 4 summarizes this data for warp 1 and warp 2. For the two algorithm Table 4.a. and shows the number of mismatches for a given utterance and speaker. In Table 4.b. the differences between the two algorithms were computed. Clearly, warp 1 and warp 2 perform differently for different utterances. For utterances with comparably long prevocalic frication or aspiration noises (e.g., c, g, z, t), warp 2 is inferior to warp 1, while for utterances with only short transitions or bursts (e.g., e, b, d), the reverse is true. To understand these differing characteristics of warp 1 and warp 2, consider the two different cases in Fig. 8.

Let us assume two simplified utterances, u_1 and u_2, that are characterized by a noisy (aspiration, frication) region n and a periodic vocalic region v. Let us furthermore assume that the noisy region of utterance 1, n_1, is much longer than that of utterance 2, n_2 (such as c, g, z compared to b, d, e). The resulting warping plane is depicted in Fig. 8.a.) A token of the class of utterance u_1 is used as an unknown test token (x-axis). As reference, tokens of type utterance 1 or utterance 2 can be used. The recognition task is to discriminate between these reference cases and select the appropriate token of type utterance 1 as the best match. For simplicity we assume here that noise will match best with noise and vocalic parts with vocalic parts, such that for the two different reference types, u_1 and u_2, dynamic programming will provide the optimum paths, p_1

and p_2. The subsequent recognition decision will choose the lower overall dissimilarity score accumulated over p_1 or p_2, respectively. Due to the properties of a speech representation based on spectral information only, distances between two noisy speech segments will be generally higher than distances between two vocalic parts (same vowel). Denoting the distances between two noisy samples as d_n and between two vocalic regions as d_v, the following overall dissimilarity scores will be obtained. Using warp 1

$$D_{w_1}(u_1,u_1) = n_1 d_n + v_1 d_v$$
$$D_{w_1}(u_1,u_2) = n_1 d_n + v_1 d_v \qquad (5.1)$$

Thus for these simplified utterances the result would be identical. For less idealized utterances the outcome will depend on the "goodness" of the distances d_v and d_n. Using warp 2 the following dissimilarity scores are obtained.

$$D_{w_2}(u_1,u_1) = (n_1+n_1)d_n + (v_1+v_1)d_v$$
$$D_{w_2}(u_1,u_2) = (n_1+n_2)d_n + (v_1+v_2)d_v \qquad (5.2)$$

Using the illustration in Fig.8, this can be written as

$$D_{w_2}(u_1,u_1) = 2n_1 d_n + 2v_1 d_v \qquad (5.3)$$
$$D_{w_2}(u_1,u_2) = 2n_1 d_n + 2v_1 d_v + (n_1-n_2)(d_v-d_n) \qquad (5.4)$$

where $n_1 > n_2$;

Thus, if we assumed d_v and d_n to be equal, the right hand side of equations (5.3) and (5.4) would be equal For $d_n > d_v$, however, $D_{w_2}(u_1,u_2) < D_{w_2}(u_1,u_1)$ and, consequently, the decision rule is more likely to choose the improper reference token for its recognition and therefore yield the confusions observed for the utterances c,g,z,etc.

The second case to be considered here is when the unknown to be recognized belongs to the group of utterances in which the vocalic part is preceded only by a short transitory region and/or a short or no burst of noise, such as b,d,e,etc. For this case, an utterance of type u_2 is matched with reference tokens of type u_1 and u_2 (Fig. 8.b.). Again assuming the simplified reference utterances u_1 and u_2, the overall dissimilarity scores for the recognition would be as follows;

For warp 1:

$$D_{w_1}(u_2,u_1) = n_2 d_n + v_2 d_v$$
$$D_{w_1}(u_2,u_2) = n_2 d_n + v_2 d_v$$

and for warp 2:

$$D_{w_2}(u_2,u_2) = 2n_2 d_n + 2v_2 d_v$$
$$D_{w_2}(u_2,u_1) = 2n_2 d_n + 2v_2 d_v - (n_1-n_2)(d_n-d_v)$$

where $n_1 > n_2$;

Again the two cases provide the same weighting conditions for an equivalent treatment of both paths in warp 1. Warp 2, however, provides different weighting conditions. Since $n_1 > n_2$ and $d_n > d_v$, $D_{w_2}(u_2,u_1) \gg D_{w_2}(u_2,u_2)$. Thus the correct token u_2 will be more likely to be chosen in this case, which explains the superiority of warp 2 for this type of utterance. Fig.9 through 12 illustrate these properties. In this case warp 1 correctly recognized the utterance "G" as "G" while warp 2 confused it with "B". Fig.9 and 10 show the search paths from both algorithms matching "G" with "B" and "G" with "G". In Fig.11 and 12 the cumulative distance along that path normalized for number of distances and weights is shown. It can be seen that for the G-G match, the disproportionality of the distances in the noise and the vocalic regions causes warp 2 to compute a higher dissimilarity score than warp 1 which in this case led to the observed confusion.

Summarizing these properties, it can be seen that warp 2 has the property of actually providing different weighting conditions if the values of the distances over segments of speech vary significantly. When comparing two such matches the one with the shorter paths through the areas of higher distances will be favored. As we have seen in certain cases, this is a desirable behavior leading to the correct recognition, while in other cases it causes confusion. Warp 1 does not have these properties, as we have seen. Alternatively, the outcome of the warp often times is adversely affected by the possibility of skipping frames and hence disregrading important transitory information. One possibility to counteract this deficiency is to select the shorter utterance in a match to be used as reference to discourage from using a steep (slope = 2) path as has been recently suggested by Das[18]. Informal experimentation with this method, however, have not yielded better results for our vocabularies.

5.2 Experiment II: Relaxation of the Boundary Constraints

The results of Experiment II are displayed in Tables 5.a. and 5.b. Wilcoxon paired comparison ranking was performed. The following ranking of "goodness" was obtained. For the digits V_1:

$$1 = 5 = 4 \overset{.01}{>} 2 \overset{.33}{>} 3$$

For the confusables V_2:

$$1 = 5 = 4 \overset{.0001}{>} 2 \overset{.0001}{>} 3$$

where the numbers represent the method number, "=" denotes equivalence and ">" superiority at significance level p (indicated by the superscript). In accordance to our previous considerations, the results indicate, that in particular for the confusables, method 2 and 3 suffer under the properties of the vocabulary. While in some cases slight inaccuracies in endpoint detection can be accounted for with this method, a greater amount of confusions is made possible by allowing for the loss of important information in the beginning of the utterance. For the digit vocabularies no significant improvements were found for either of the investigated methods. As we shall see in the next section, most recognition errors are caused for this vocabulary by inaccurate endpoint detection. Some of these endpoint errors, however, include severe loss of accurate information, for which the present methods could not compensate. In such cases, word spotting techniques that recognize partial equivalence between two utterances might prove more useful[15,23].

5.3 Experiment III: Adjustment Window

As has been noted before, it is clear from Fig. 7 that the computational saving is directly dependent on the length of the two utterances to be matched. In the limiting case, one utterance exceeds the length of the other by a factor of two, the search space spun by the slope constraints reduces to zero and the warp can be aborted. We have thus generated histograms of duration of words for three different vocabularies (V_1, V_2, and the alphadigits). Fig. 13 through 15 shows three typical examples for speaker FA. In order to obtain an estimate on the computational savings for different values of t, the expected average saving has been computed, assuming all combinations of tokens for a vocabulary and speaker have been used, as is the case in these experiments. The saving is assumed to be proportional to the number of grid points in the search space that were discarded by the restriction imposed by the window. Fig. 16 through 18 expresses these results in percentage saving. Zero percent saving implies the entire parallelogram search space had to be computed, while 100% saving means, no computation was performed. Even for linear time normalization (t=0), the minimum computation needed is for all the points on the diagonal and hence the saving will always be somewhat below 100%.

Fig. 19 and 20 display the recognition results for two vocabularies (V_1, V_2) for the values of t used (0, 3, 5, 8). In agreement with[11], the superiority of dynamic programming (tolerance >0) over linear time normalization (t = 0) can be seen here. It is displayed here for the purpose of comparison. Increasing t generally improves recognition results, up to t = 5. For the confusables (V_2), recognition accuracy even reaches its highest value for five of the eight speakers (SW, DS, FA, RP, JL) for t = 5. Moreover, for speakers MS and GG the improvements gained by using t > 5 are marginal. In case of the digit vocabulary, for six of the eight speakers recognition rates do not improve or even degrade, when t is increased beyond the value 8. For five speakers, t = 5 is even sufficient to yield nearly equivalent (degradation < .25%) results. For the speakers MA and GG only, significant degradation can be seen when the search path is restricted by the window function. The reason for this behavior is due to grave begin-end time errors (missing noise portions for "three" or "six"). Allowing for the search path to grow into the corners of the parallelogram increases the likelihood that the path might allow one utterance to "catch up" with the other under the presence of incorrect endpoints. The present results reflect this property strongly, because of the permutative way of matching all data sets in our data base in these experiments, i.e., one incorrect endpoint might cause several errors. For a practical recognition system, this problem would be eliminated by means of alternate techniques such as the word spotting methods mentioned previously, or alternatively, by rejecting the entire match when a certain threshold of dissimilarity is reached and asking the user to repeat, etc.

Comparing the results for the digit vocabulary and the confusables, it is helpful to see that the nature of the problems causing confusion is different. Most problems for the digit vocabulary are due to errors in the endpoint detection, while recognition results of the confusables are mostly affected by the genuine recognition problem, i.e., to derive a discriminatory decision from a set of highly similar speech signals. As such, it can be understood that in the latter case (V_2) results can actually be improved often by restricting the search path, since (assuming no significant endpoint detection errors) linguistically not meaningful search paths are inhibited. In conclusion, for use of an alpha-digit isolated word recognition system, a window constraint of tolerance five frames, i.e., ±50 msec deviation from the diagonal in the search space, can be suggested. From Fig. 16 through 18, we see that this window constraint leads to computational savings in the range of 50% to 70%. The usage of a 50 msec window can be interpreted as correcting matches on a frame by frame basis dynamically to lead ahead of or lag behind a linearly compressed or expanded mapping of two utterances. This implies that a segment in an utterance read in isolation is unlikely to vary in duration by more than 50 msecs. For isolated and possibly connected word recognition systems we believe that this result can be generalized onto other vocabularies.

6. Summary and Conclusions

In this paper we have investigated several nonlinear warping methods proposed in the literature in order to optimize both recognition accuracy and computational efficiency. These investigations were conducted in view of vocabularies with varying degrees of difficulty of discrimination.

6.1 Warping Algorithm

The asymmetric dynamic programming algorithm proposed by Itakura was seen to be the best solution for a vocabulary for which recognition depended critically on the discrimination between noisy, aperiodic, transitory regions of the speech signal (such as vocabulary V_2). We have discussed the defficiencies of the Itakura algorithm and of the symmetric Sakoe & Chiba algorithm in detail and explained the reasons for various algorithms to perform differently when different vocabularies were used. These defficiencies are fairly subtle, but appear with significance in highly ambiguous vocabularies as the ones we studied. Choosing between these algorithms we have decided in favor of the asymmetric Itakura algorithm with practical considerations in mind, namely to enable the extension to connected speech[2] as provided by an asymmetric algorithm. Some of the more fundamental problems of dynamic programming are the fact that all segments receive equal treatment although the perceptual cues encoded in the signal are of differing nature. In this fashion, present methods almost exclusively rely on the spectral information. For vocalic regions this is a sufficiently reliable description, but for consonantal regions, cues such as noise energy, duration,and formant transitions are neglected or "warped away". It is our hope that feature based knowledge, implemented either within the framework of dynamic programming or as a post processor, might in future research greatly enhance reliability and recognition accuracy of isolated and connected speech recognition systems.

6.2 Relaxing Boundary Constraints

All methods tested have been seen as not to improve recognition results significantly. While relaxing the boundary constraints in some cases can account for endpoint detection errors by dynamically choosing the "best" begin or endpoints within a certain tolerance, it provides in other cases an additional source of errors by allowing the algorithm to omit important short segments at the boundaries (as is the case for some utterances in the alpha-digit vocabulary).

6.3 Restricting of the Search Space by an Adjustment Window

The results indicate that a window that restricts a dynamic programming search path to deviate from a linear match only by up to 50 msecs is an optimal choice for isolated word recognition. This window constraint not only saves up to 70% computation at no loss of recognition accuracy, but even improves recognition accuracy in many cases by virtue of restricting the search to linguistically meaningful matches.[2]

[2] It is important to note that the recognition accuracy of a system depends on the nature of vocabulary and the Speaker. Figures 19 and 20 illustrate this point, especially for the digit vocabulary given in Fig. 19. The realizable error rate varies from 0.2% to 5% depending on the speaker. Therefore, comparison of recognition systems performance based on error rate alone is not correct, although it is often seen in the literature.

Acknowledgement

The authors wish to thank Dr. Raj Reddy for his encouraging advise and support that made this study possible. Also special thanks to Neeraja Krishnan for her programming support and her patient assistance in conducting the experiments for this paper. Finally, we are grateful to Jan Asbury for her help in preparing this document.

7. Tables

V1	V2
ONE	B
TWO	C
THREE	D
FOUR	E
FIVE	G
SIX	P
SEVEN	T
EIGHT	V
NINE	Z
ZERO	

Table 1 Vocabularies V1 and V2 have been used separately for testing in subsequent experiments.

Table 2a.) Recognition rates obtained using three warping algorithms (digit vocabulary V1).

	Itakura	Ssym	Sasym
fa:	99.89	100.00	100.00
ms:	97.56	97.67	97.45
ma:	96.34	96.56	95.11
rp:	100.00	100.00	99.78
jl:	96.89	96.67	96.89
ds:	95.34	95.34	95.11
sw:	97.45	97.45	96.67
gg:	99.89	99.78	99.89

Table 2b.) Recognition rates obtained using three warping algorithms (confusable vocabulary V2).

	Itakura	Ssym	Sasym
fa:	68.77	67.28	67.04
ms:	61.48	60.99	59.14
ma:	48.77	45.80	44.82
rp:	77.28	78.52	77.53
jl:	65.06	64.07	63.46
ds:	69.63	69.14	69.87
sw:	44.44	42.72	42.47
gg:	43.70	41.23	39.88

Itakura

test \ reference	e	v	b	p	d	t	g	c	z	Tot
e	37		15	2	27					44
v		55	3		1	9		3	10	26
b	17	3	30	2	27	2				51
p		2		40		33	5	1		41
d	24	5	19	1	29	2	1			52
t		3	1	5		68	2	2		13
g			3	6		9	64			17
c								81		
z		2							79	2
	e	v	b	p	d	t	g	c	z	Tot

Error Percentage (246/810) =30.37

Ssym

test \ reference	e	v	b	p	d	t	g	c	z	Tot
e	43		15		23					38
v		60	9		3	4		2	3	21
b	21	2	31		27					50
p	1	2	4	43	1	26	4			38
d	26	2	23		28	1	1			53
t		3	1	7	1	68	1			13
g	1	1	4	10		11	54			27
c		1				1		79		2
z		8							73	8
	e	v	b	p	d	t	g	c	z	Tot

Error Percentage (250/810) =30.86

Table 3 Two typical confusion matrices (speaker DS) using the two warping algorithms Warp 1 and Warp 2 over the confusable vocabulary V2.

	Itakura									Ssym								
	e	v	b	p	d	t	g	c	z	e	v	b	p	d	t	g	c	z
ds	44	26	51	41	52	13	17	0	2	38	21	50	38	53	13	27	2	8
fa	39	13	44	43	45	38	8	4	19	26	14	43	48	41	50	8	17	18
gg	40	57	67	60	48	54	41	23	66	36	55	70	62	45	55	46	39	68
jl	35	63	37	48	32	20	26	3	19	34	63	31	45	32	28	31	8	19
ma	70	39	52	42	25	62	47	38	40	70	39	49	49	27	61	50	47	47
ms	30	49	31	58	50	35	0	16	43	24	52	26	58	45	36	2	28	45
rp	10	13	43	33	42	17	20	1	5	8	19	37	28	35	18	23	2	4
sw	57	56	58	56	51	64	56	8	44	56	59	56	60	45	64	64	15	45

Table 4.a. Confusion matrices using warp1 and warp2 (V2 vocabulary) All numbers indicate number of confusions out of 810 recognitions.

	D-SCORES								
	e	v	b	p	d	t	g	c	z
ds	6	5	1	3	-1	0	-10	-2	-6
fa	13	-1	1	-5	4	-12	0	-13	1
gg	4	2	-3	-2	3	-1	-5	-16	-2
jl	1	0	6	3	0	-8	-5	-5	0
ma	0	0	3	-7	-2	1	-3	-9	-7
ms	6	-3	5	0	5	-1	-2	-12	-2
rp	2	-6	6	5	7	-1	-3	-1	1
sw	1	-3	2	-4	6	0	-8	-7	-1

Table 4.b. Difference scores between confusions of warp1 and warp2.

Table 5a.) Recognition rates obtained when the boundary constraints where relaxed according to method 1 through method 5 (digit vocabulary V1).

	skip0	skip3	skip5	vskip3	vskip5
fa:	99.89	100.00	100.00	99.89	99.89
ms:	97.56	95.34	95.33	97.67	97.67
ma:	96.34	90.67	90.56	96.89	97.22
rp:	100.00	99.33	99.22	99.89	99.89
jl:	96.89	95.34	95.22	96.89	96.89
ds:	95.34	95.34	95.34	95.34	95.34
sw:	97.45	97.34	96.67	97.56	97.56
gg:	99.89	96.34	95.89	99.78	99.78

Table 5b.) Recognition rates obtained when the boundary constraints where relaxed according to method 1 through method 5 (confusable vocabulary V2).

	skip0	skip3	skip5	vskip3	vskip5
fa:	68.77	55.19	44.44	67.78	65.19
ms:	61.48	51.73	39.13	60.25	59.51
ma:	48.77	41.73	34.20	49.13	49.38
rp:	77.28	70.37	59.75	76.17	76.91
jl:	65.06	57.16	53.21	65.56	65.06
ds:	69.63	55.31	42.10	69.01	67.28
sw:	44.44	36.05	27.90	42.59	42.84
gg:	43.70	37.78	32.47	44.44	44.32

8. Figures

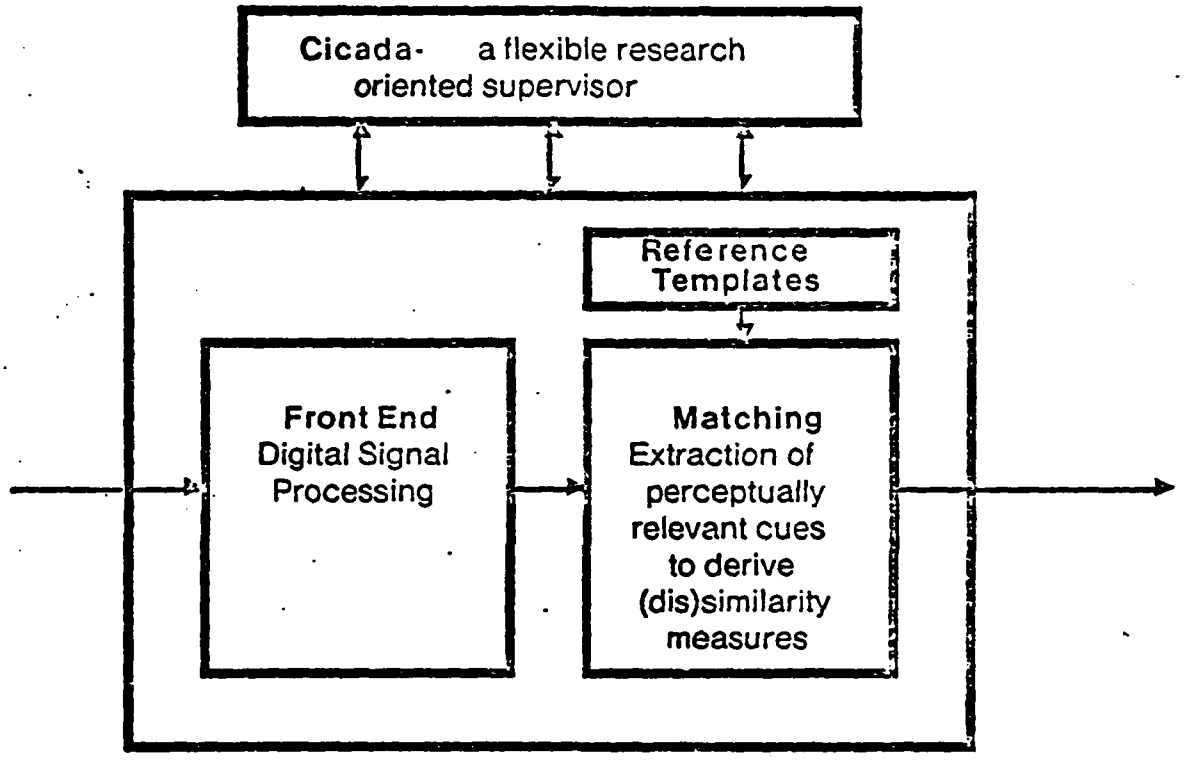


Fig. 1 Overview of the isolated word recognition system

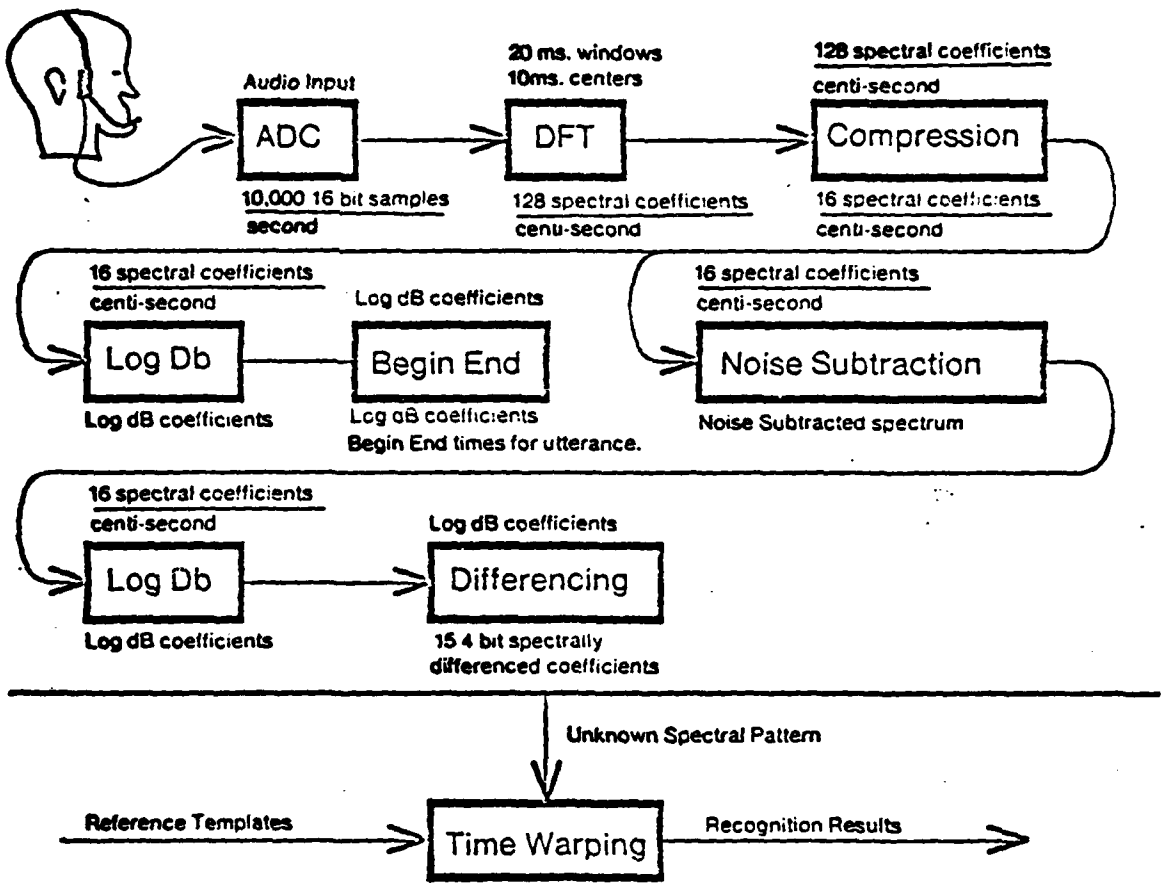


Fig.2 Diagram of the isolated word recognition system

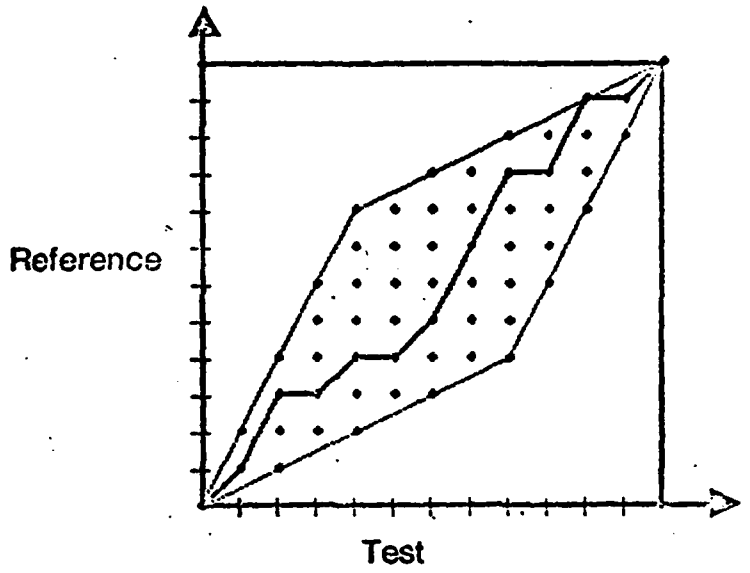
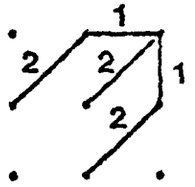


Fig.3 Dynamic programming search path aligning test and reference token

warp1
Itakura asymmetric

warp2
Sakoe & Chiba symmetric



warp3
Sakoe & Chiba asymmetric

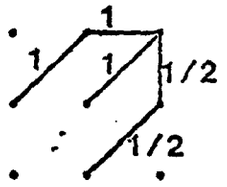


Fig.4 Three investigated warping functions

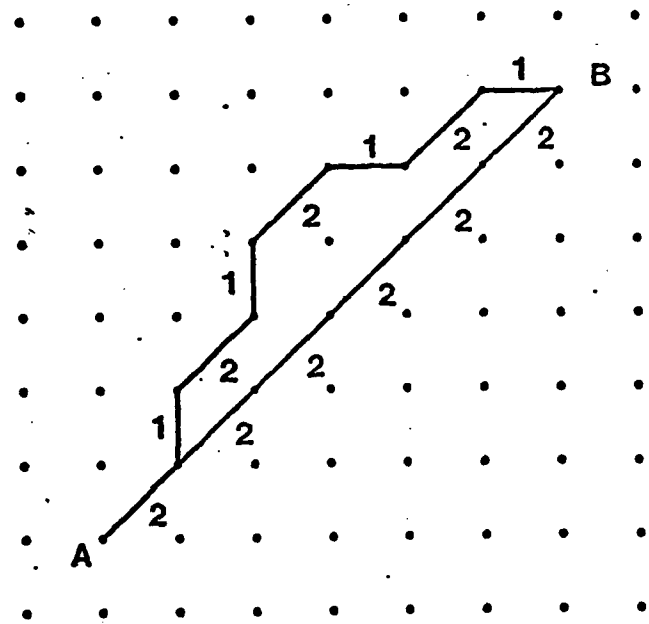
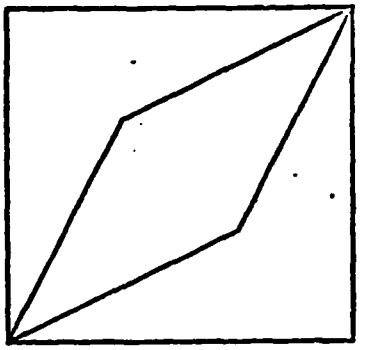
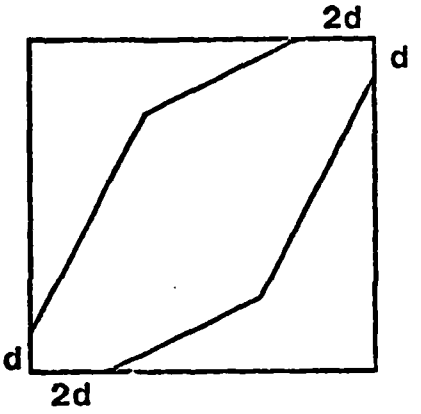


Fig.5 Weights for Symetric Warp

a.) Method1
constraint endpoints



b.) Method2 (d = 3)
Method3 (d = 5)
unconstraint endpoints in test and reference



c.) Method4 (d = 3)
Method5 (d = 5)
unconstraint endpoints in reference only

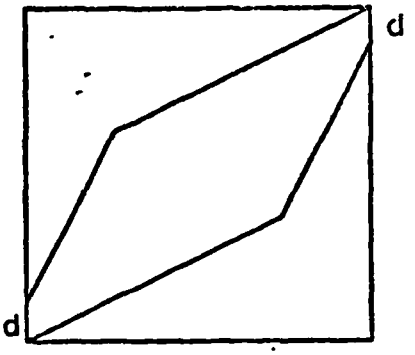


Fig.6 Methods to relax the boundary constraints to account for endpoint detection errors

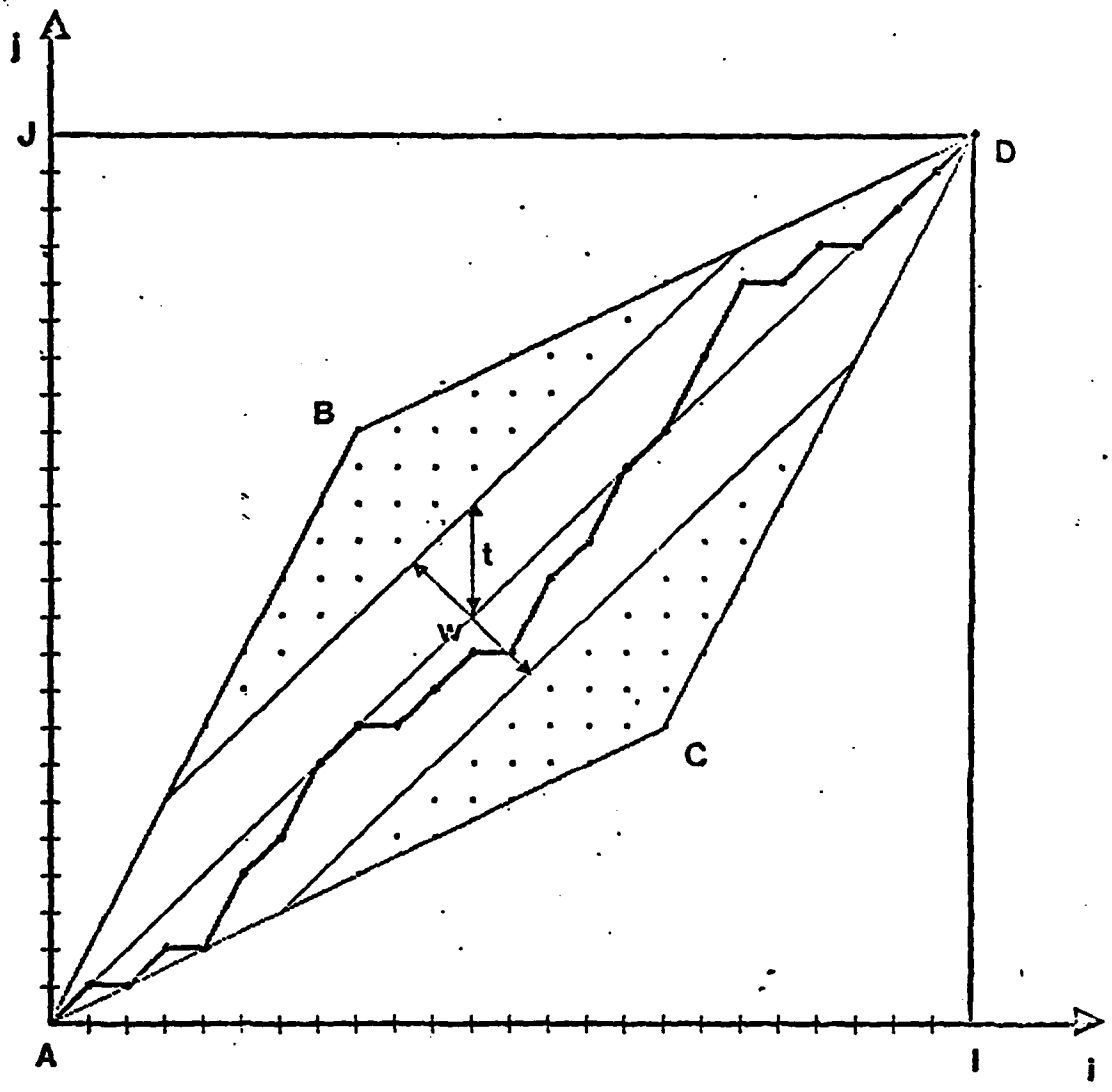


Fig.7 Restriction of the search space via an adjustment window
The dotted area indicates computational saving through the use of the window constraint. Tolerance t is used as a measure of the width as well as the saving achieved.

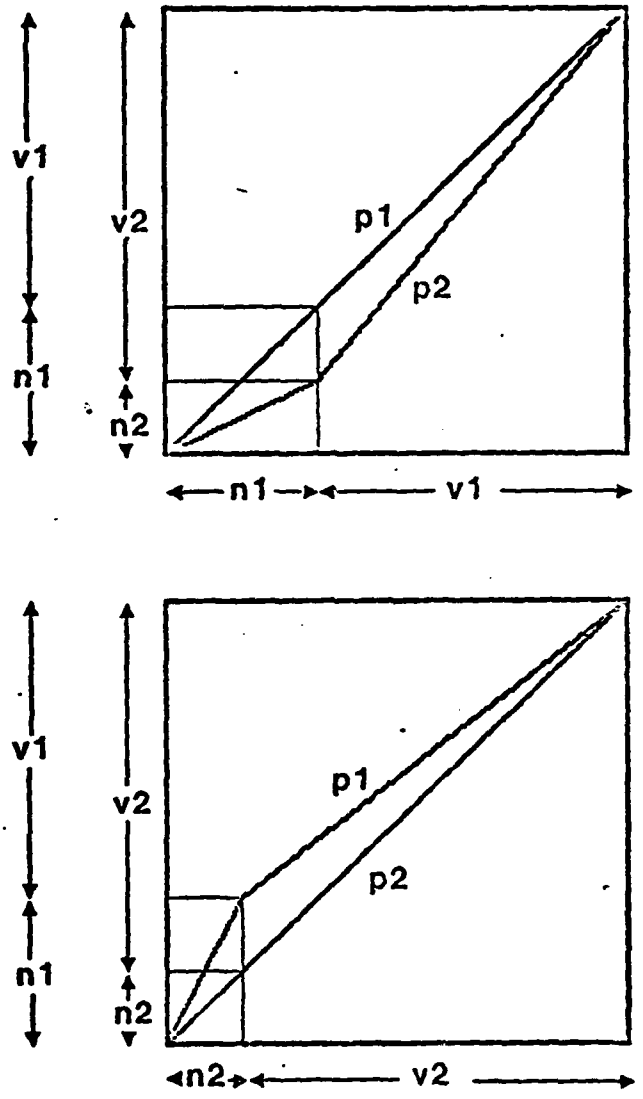


Fig.8 Properties of a symetric warping algorithm in different regions of an utterance.

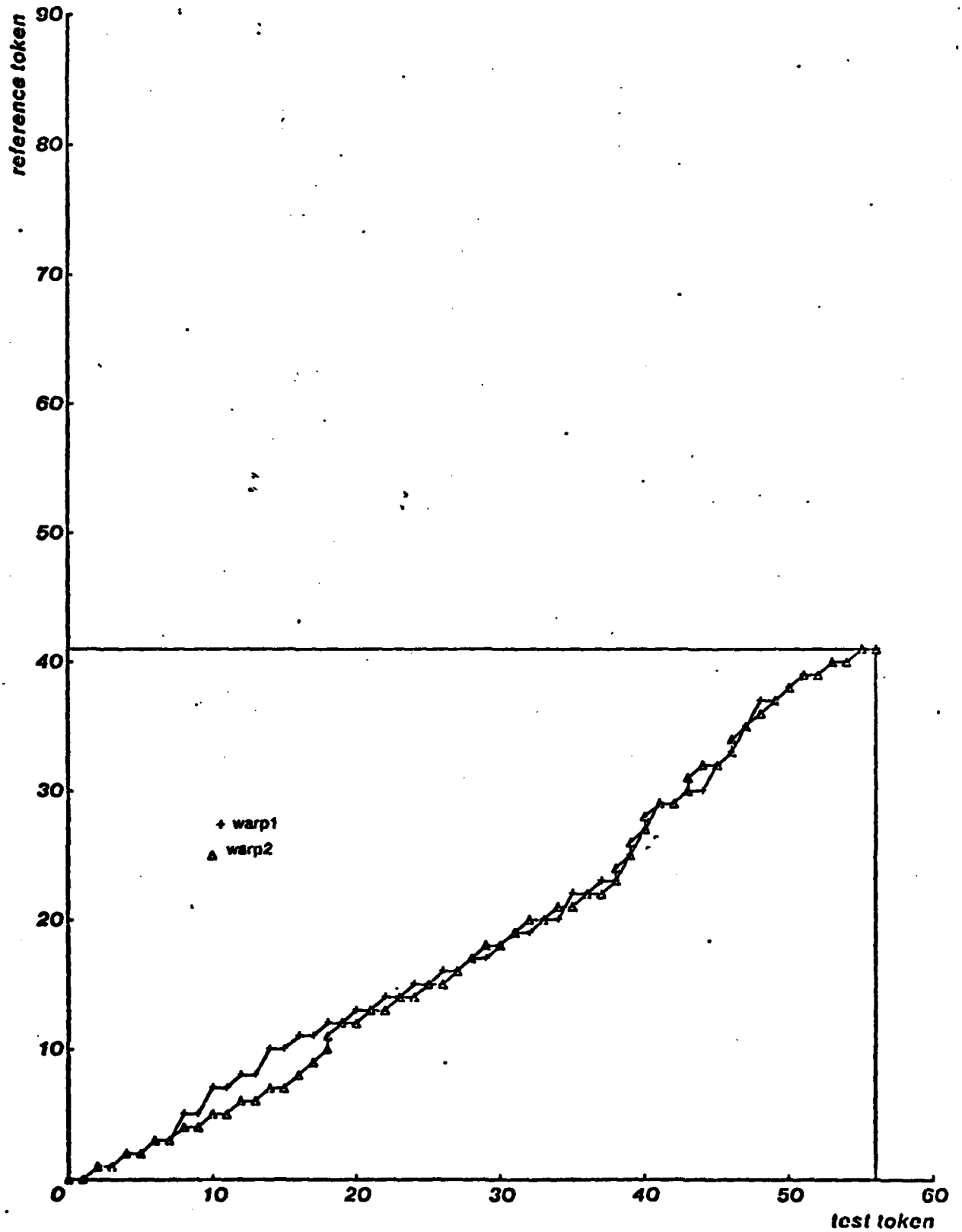


Fig.9 Warping path for the match between G and B

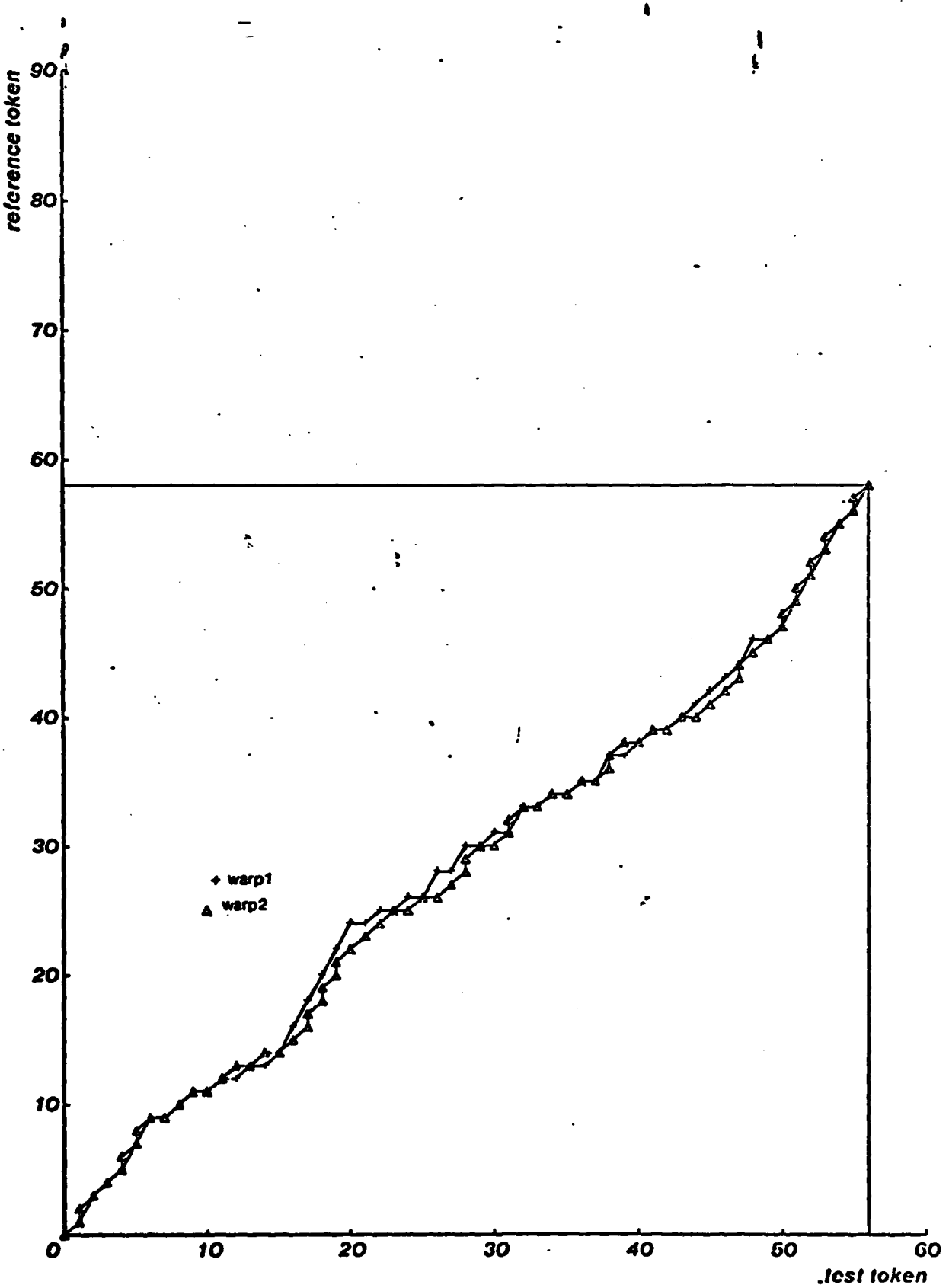


Fig. 10 Warping path for the match between G and G

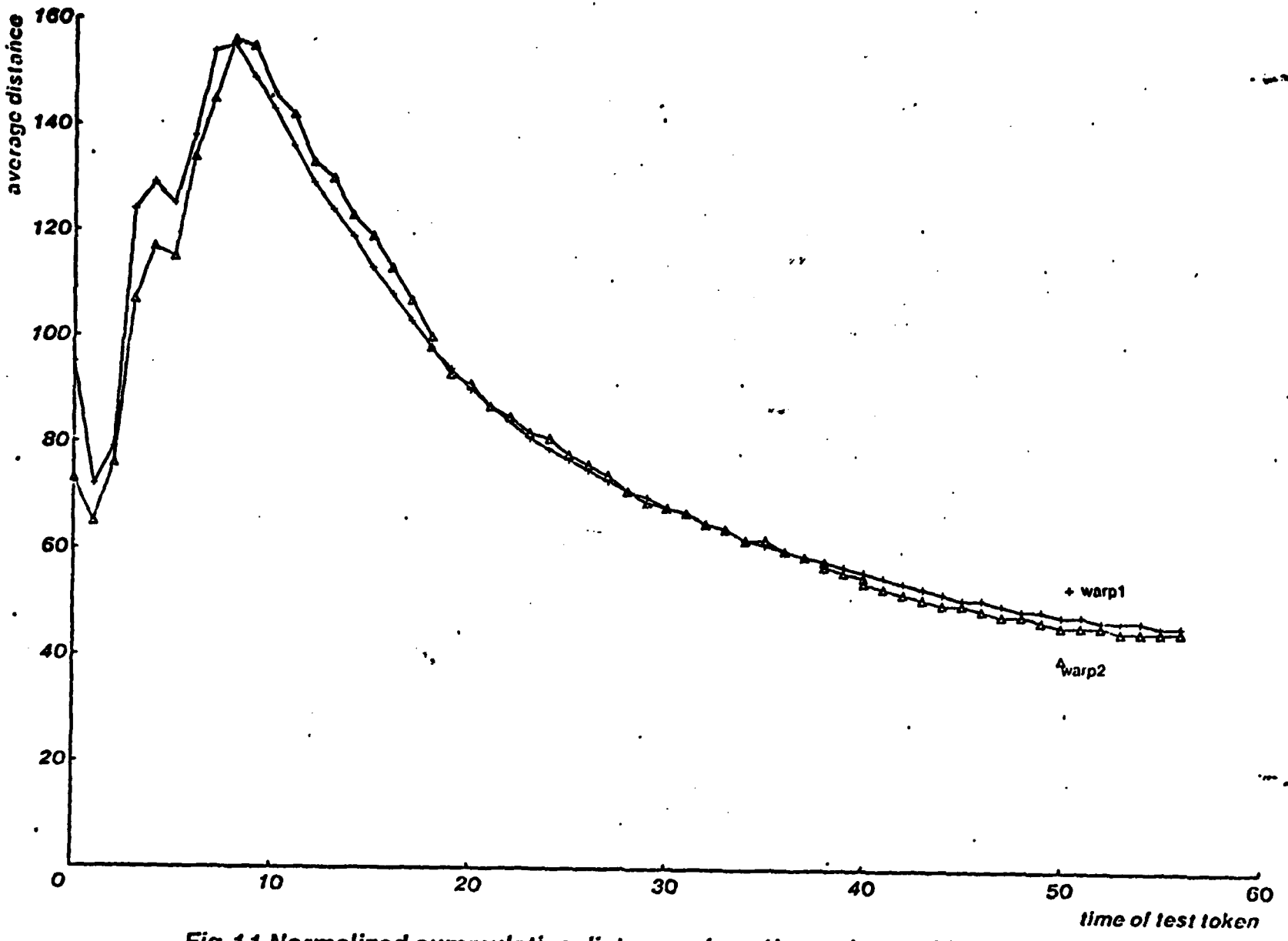


Fig.11 Normalized cummulative distance along the path matching G and B

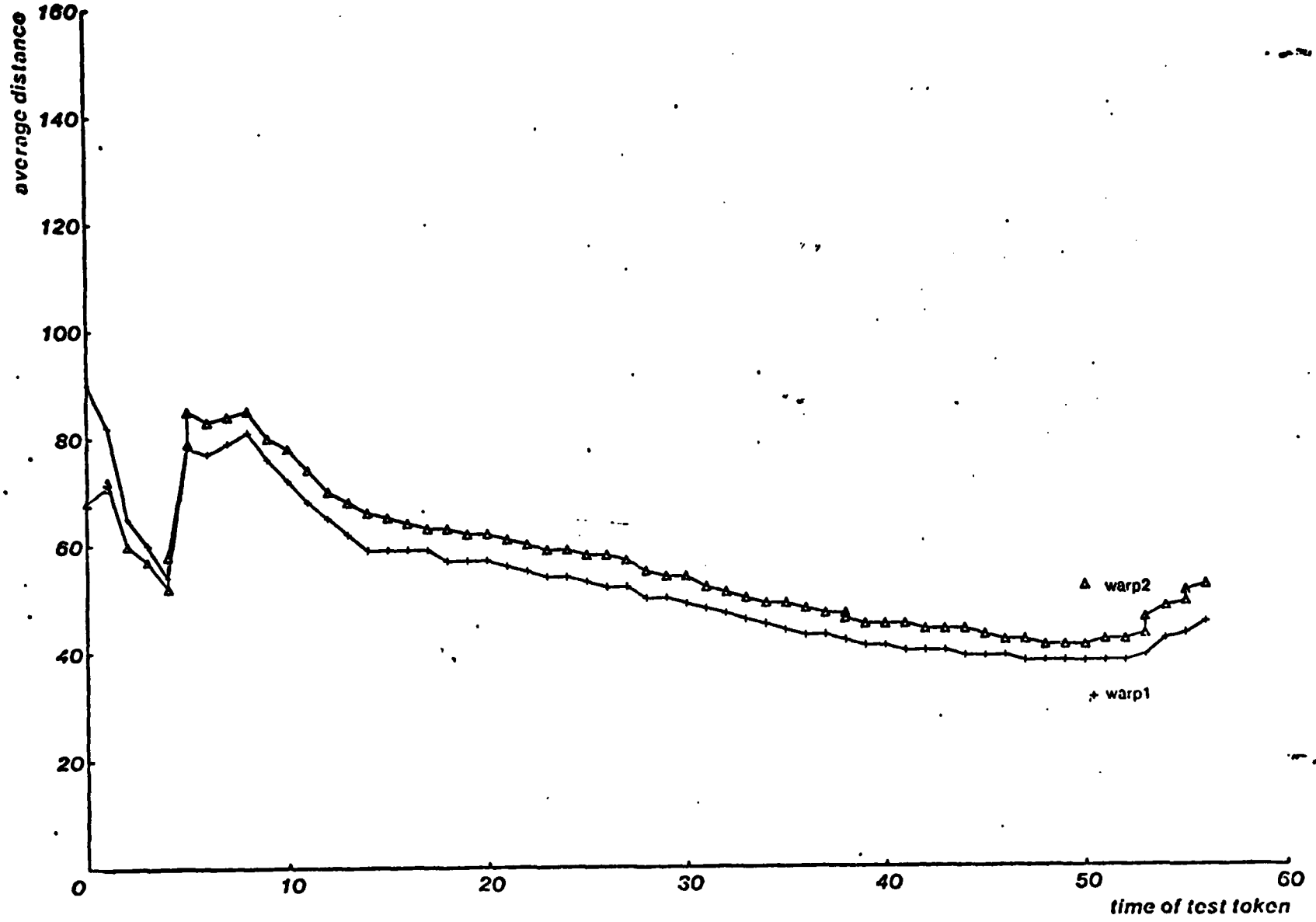


Fig. 12 Normalized cummulative distances along the path matching G and G

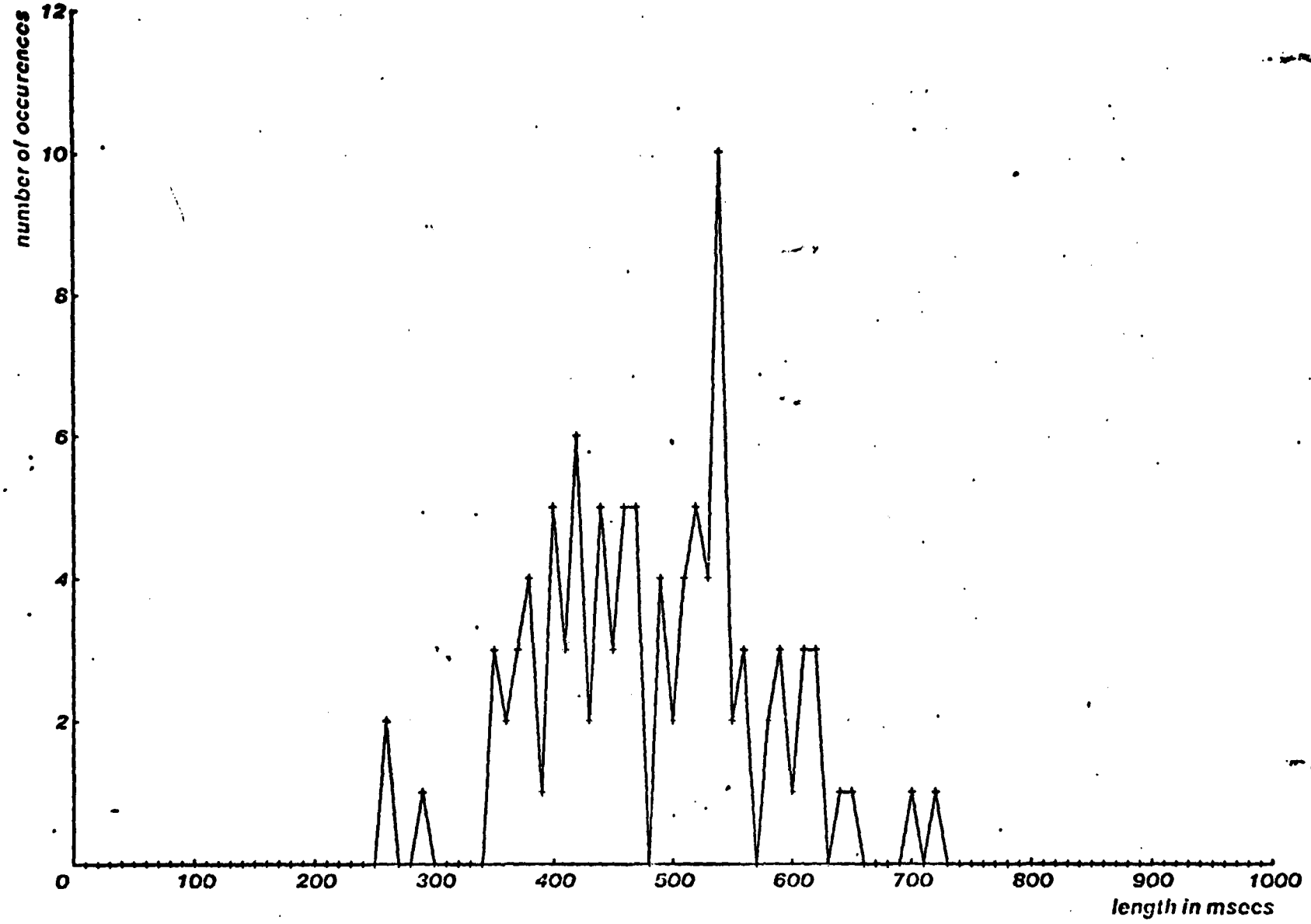


Fig. 14 Histogram of digit utterence lengths (speaker FA)

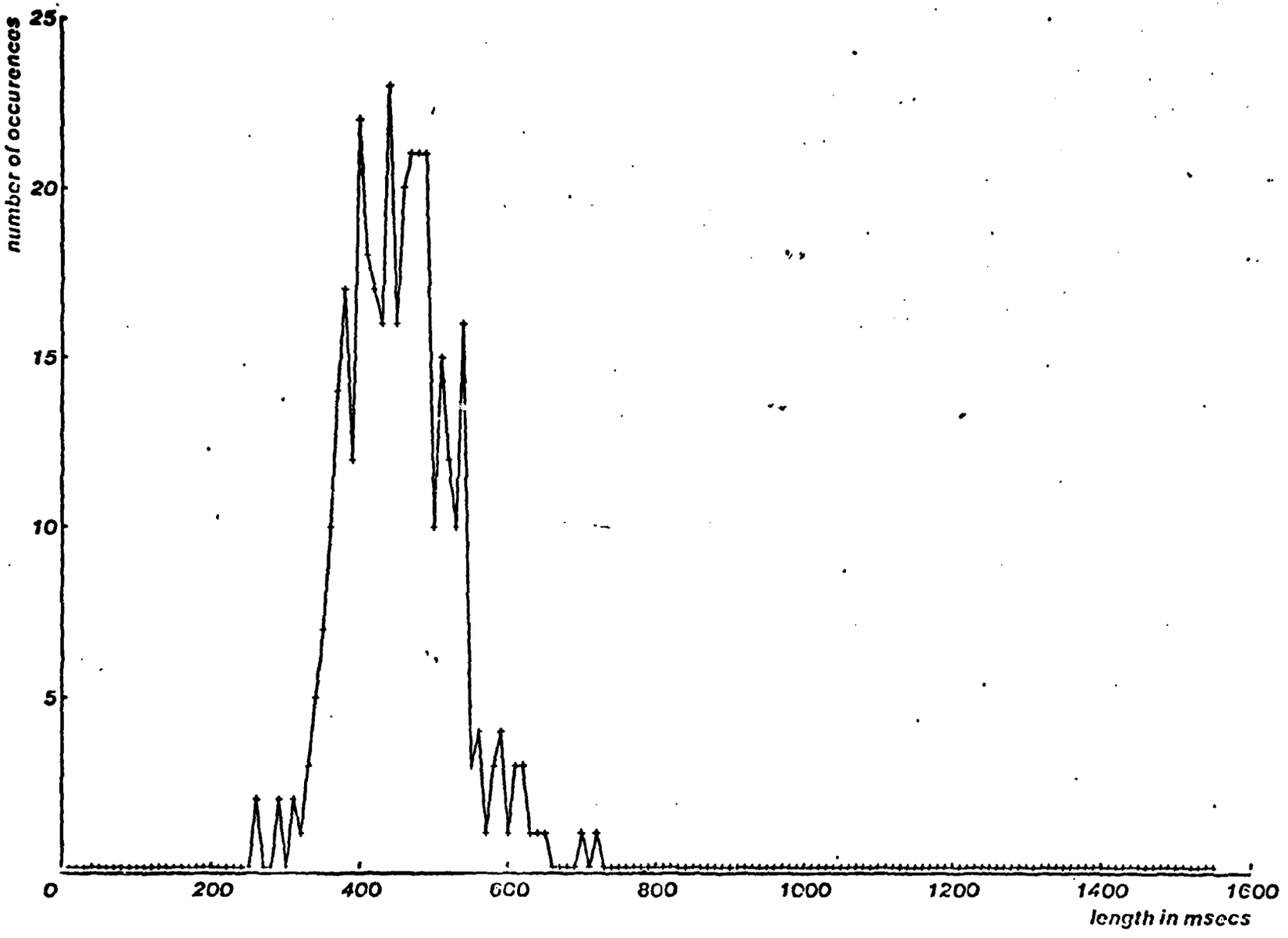


Fig.13 Histogram of alpha-digit utterence lengths (speaker FA)

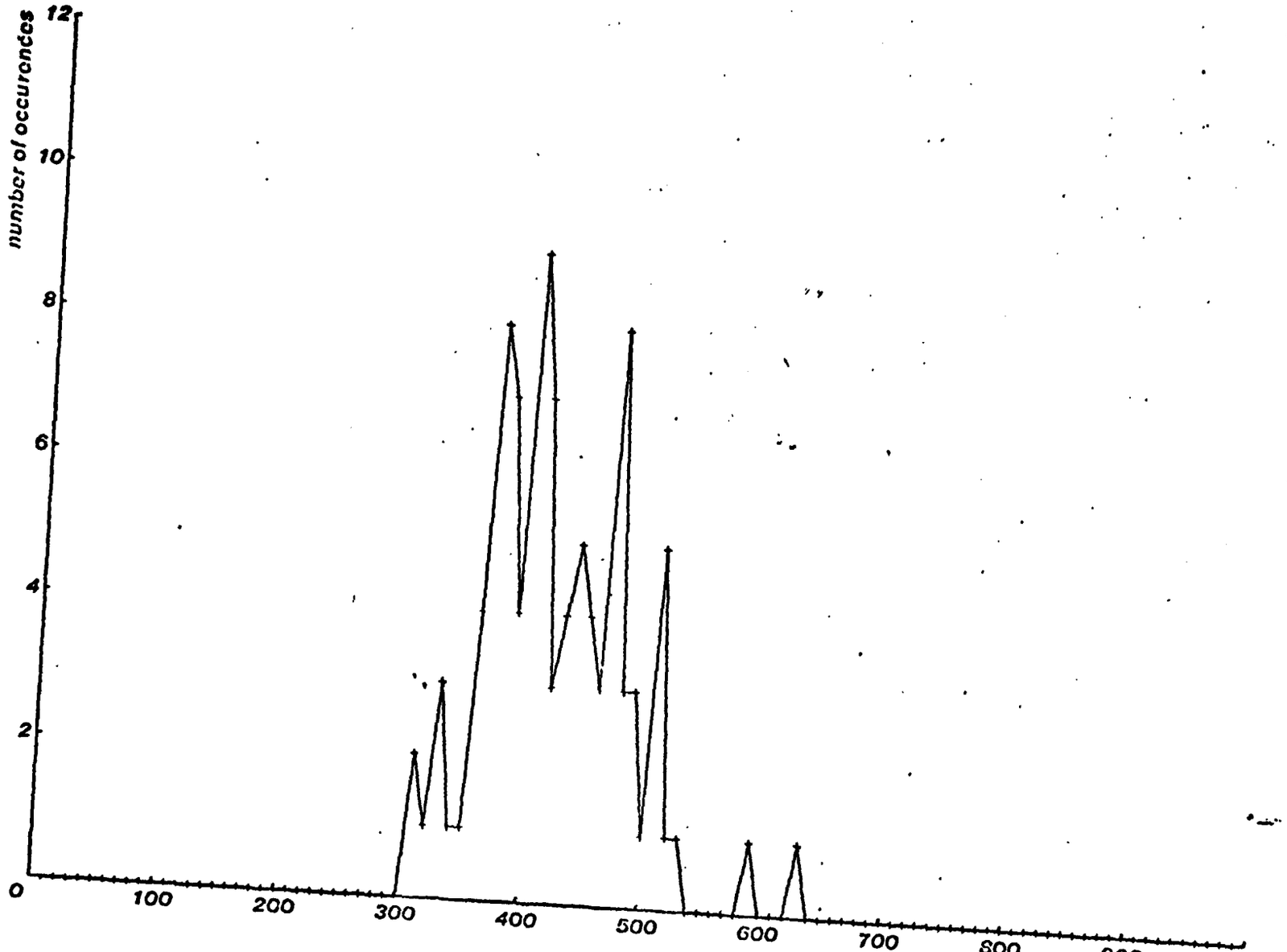


Fig. 15 Histogram of confusable utterence lengths (speaker FA)

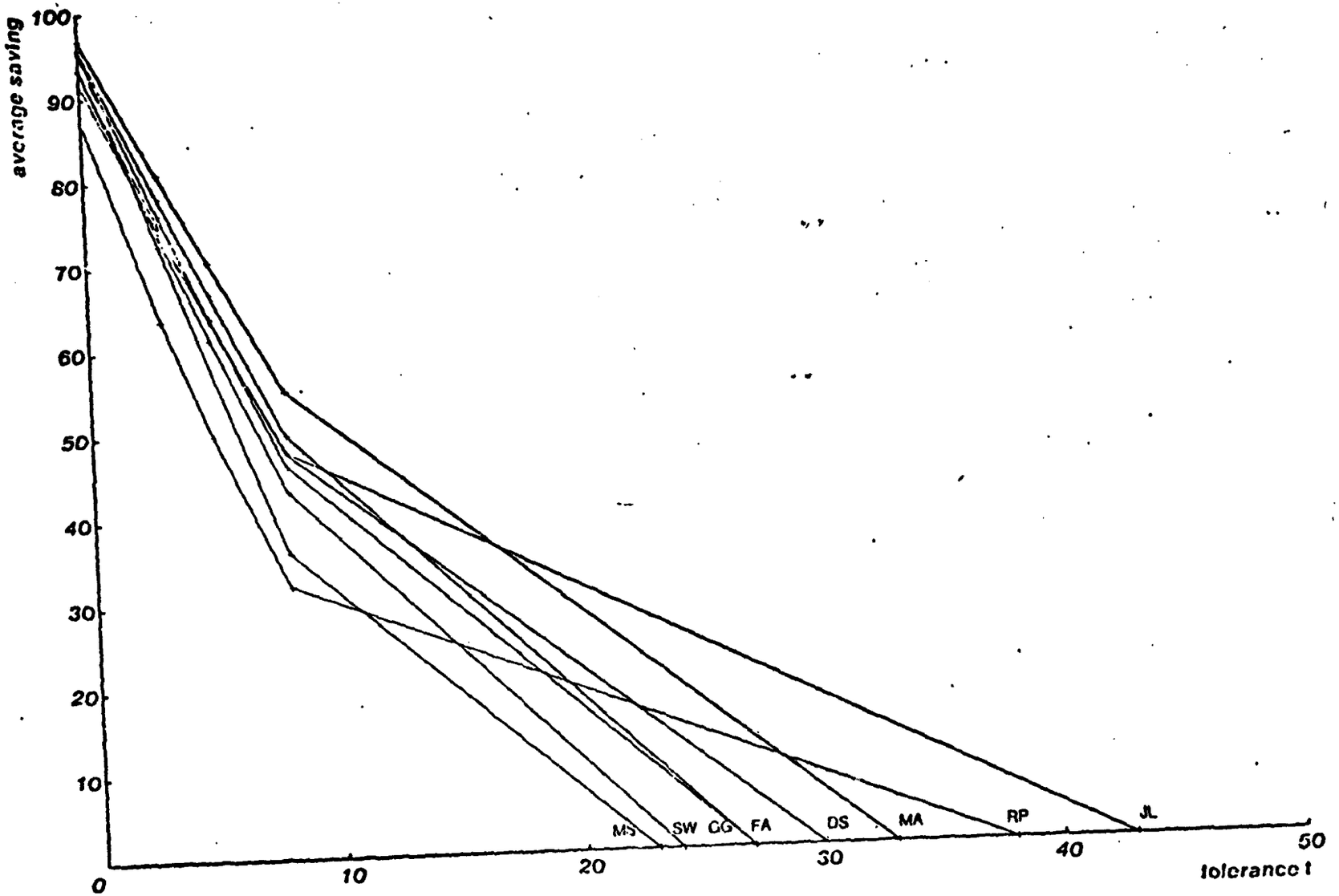


Fig.16 Computational Saving (Alpha-digit vocabulary)

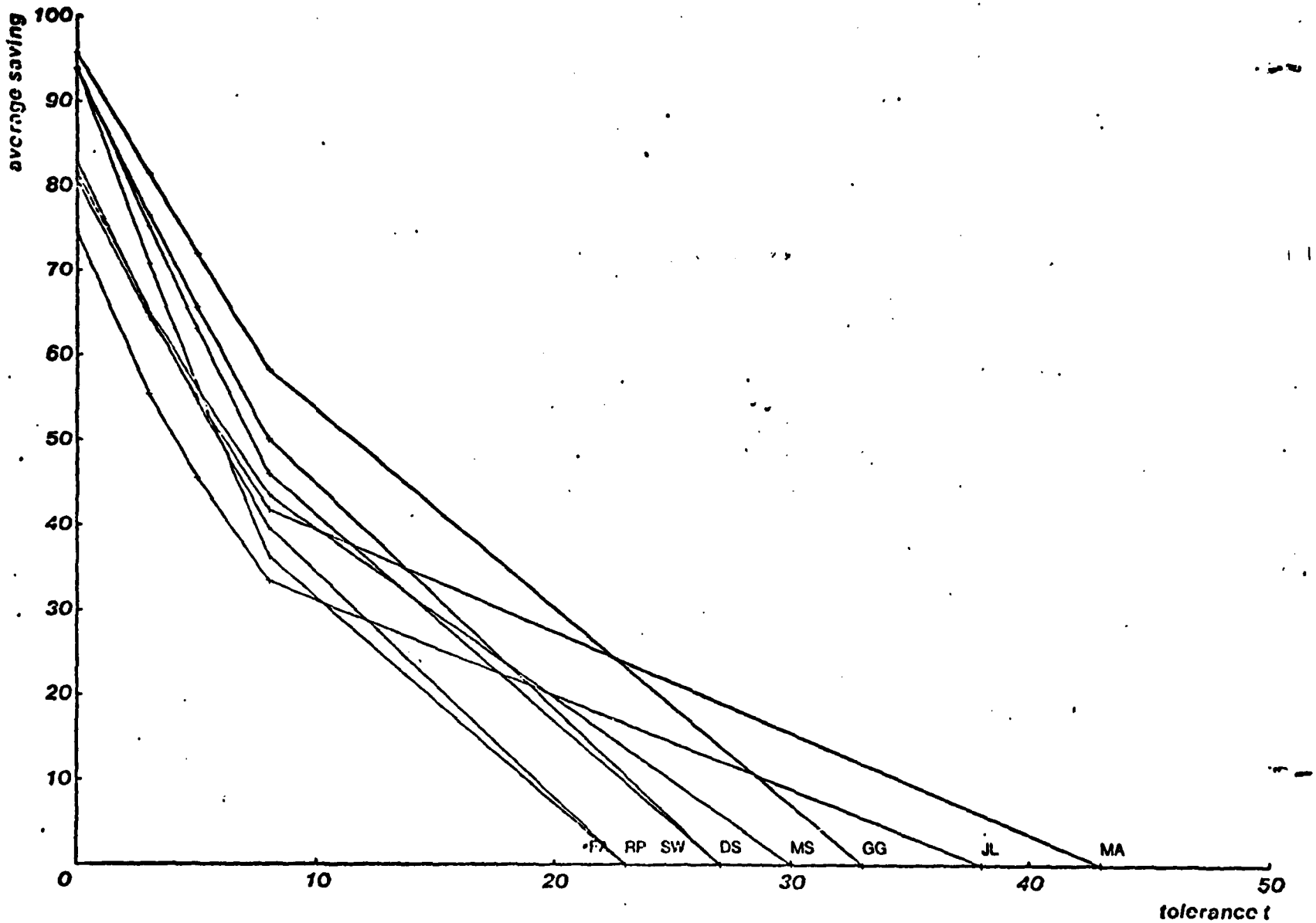


Fig.17 Computational Saving (Digit vocabulary V1)

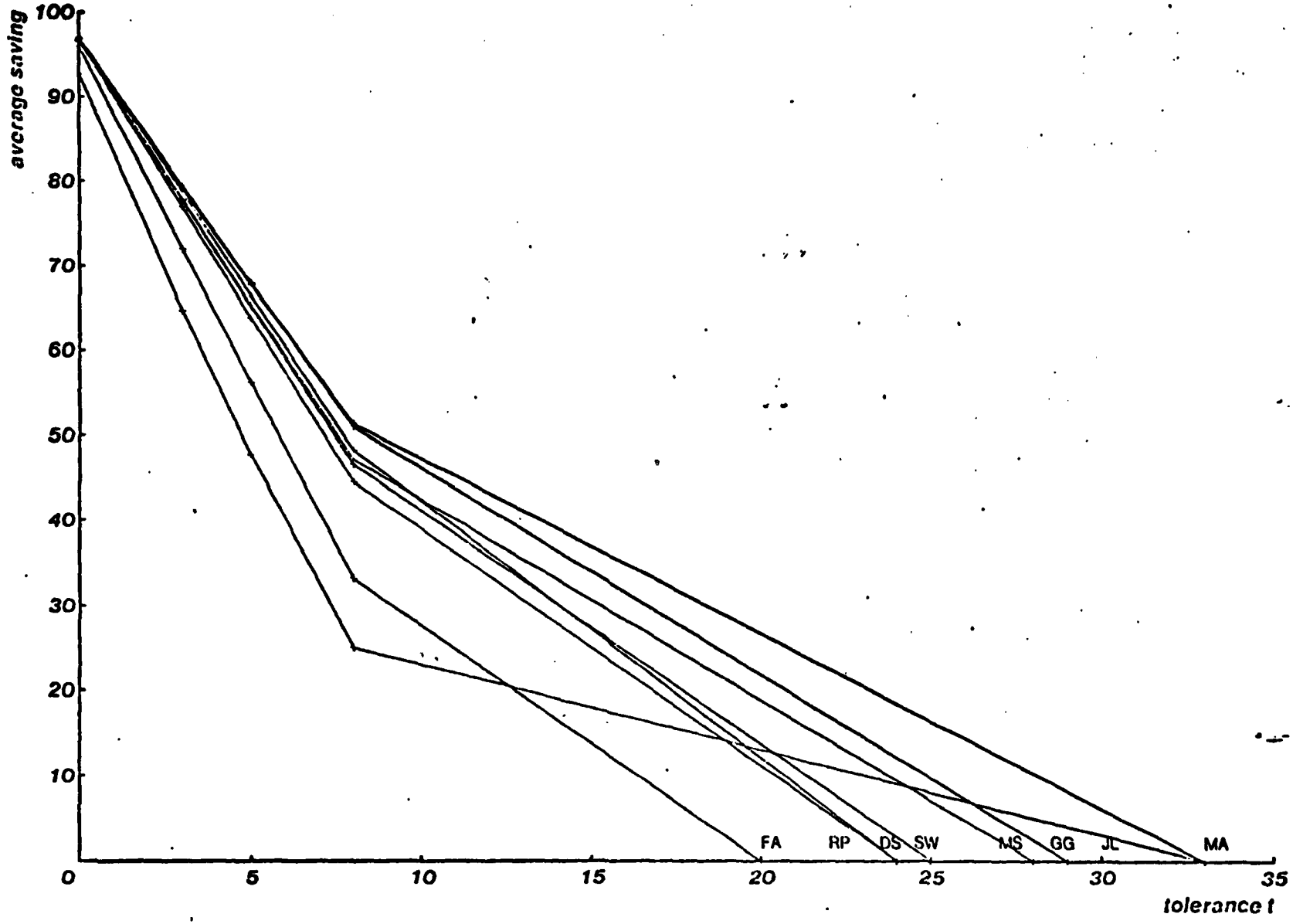


Fig.18 Computational Saving(Confusable vocabulary V2)

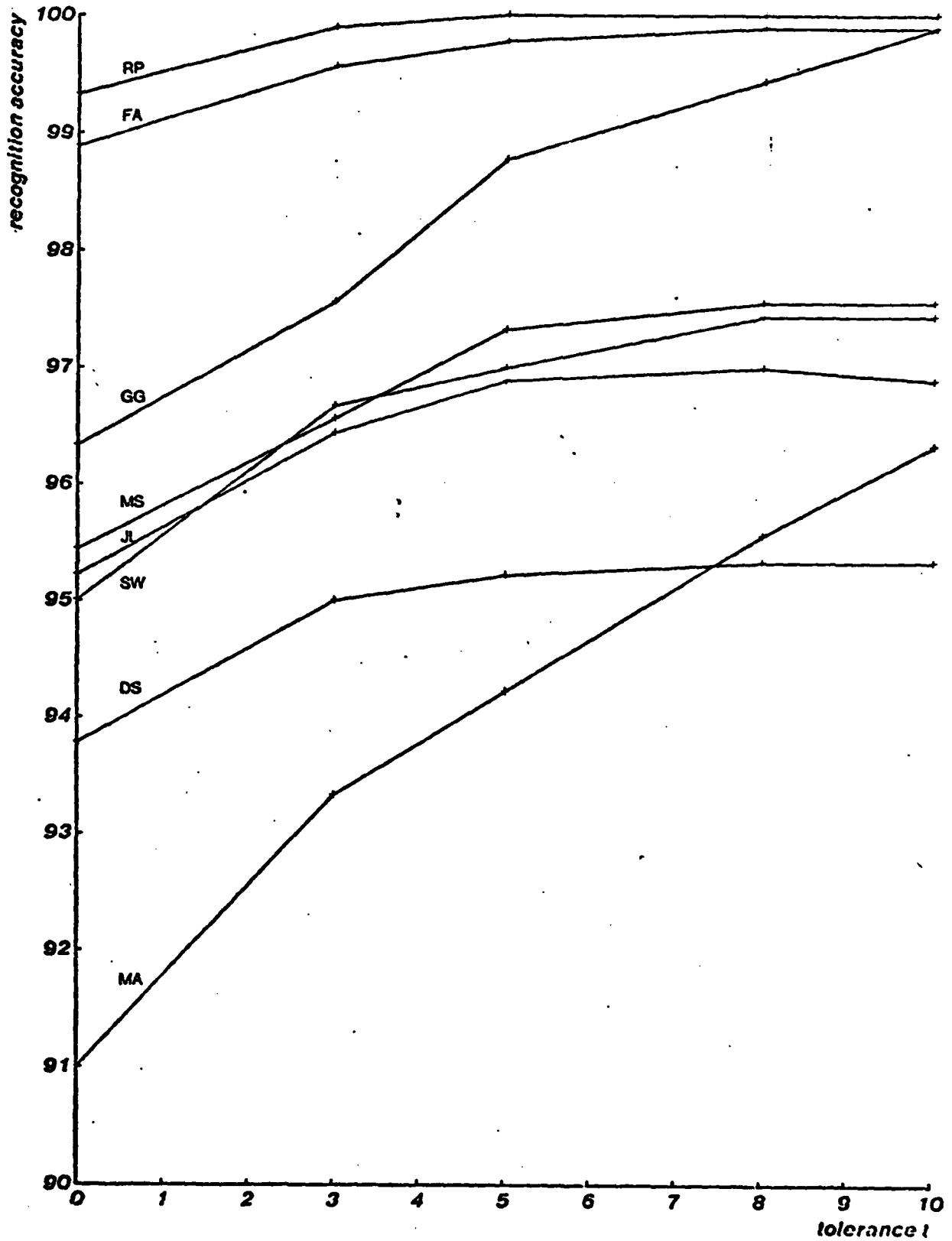


Fig. 19 Recognition accuracy for the digit vocabulary (V1)

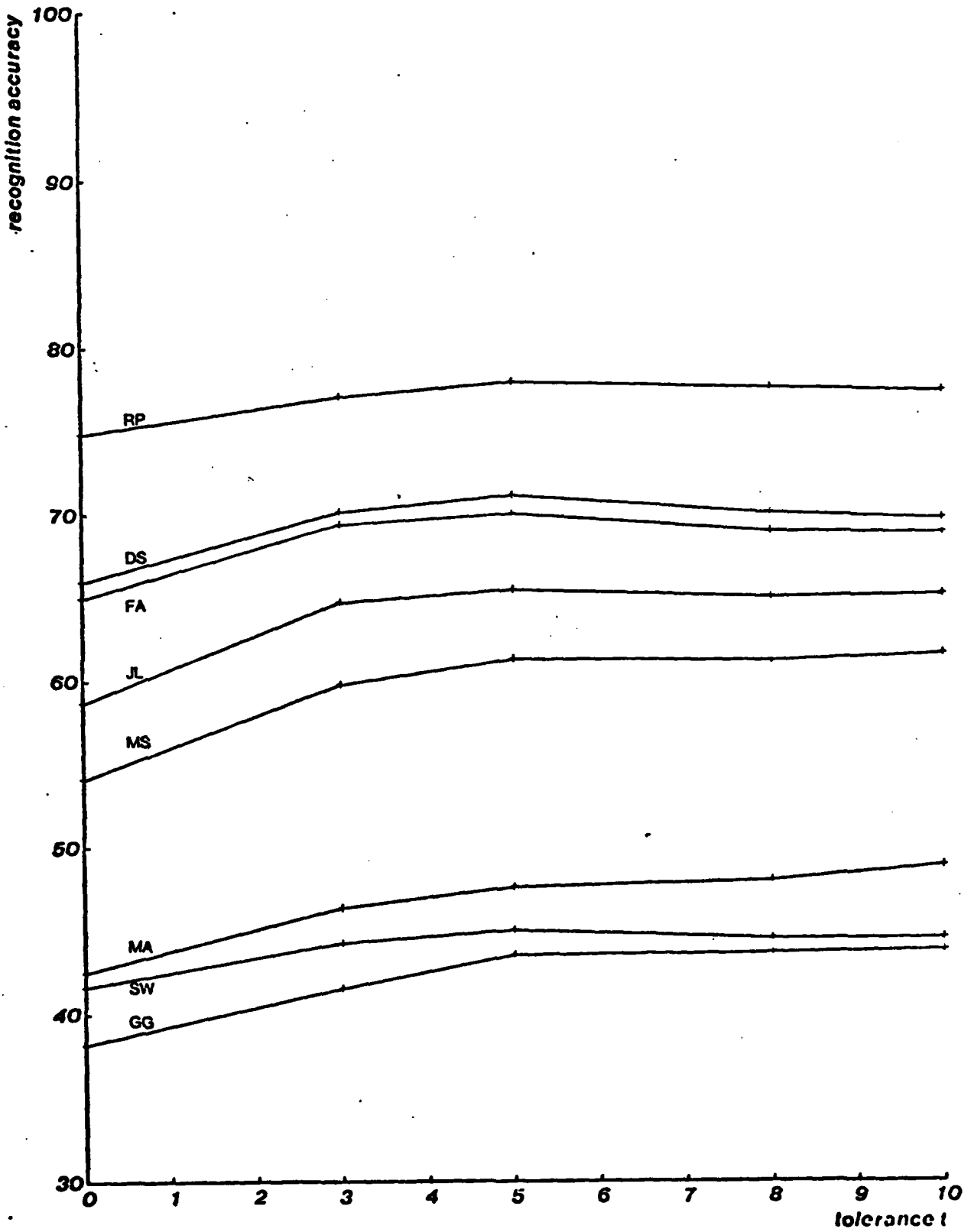


Fig.20 Recognition accuracy for the confusable vocabulary (V2)

References

1. T.Martin, "Practical Applications of Voice Input to Machines," *IEEE Transactions on Acoustics, Speech, Signal Processing*, Vol. 64, April 1976, pp. 487-501.

2. Y.Kato, "NEC Connected Speech Recognition System," Tech. report, Central Reasearch Laboratories, Nippon Electric Company, 1979.

3. D.R.Reddy, "Speech Recognition by Machine: A Review," *Proceedings IEEE*, April 1976, pp. 501-531.

4. B.T.Lowerre, *The Harpy Speech Recognition System*, PhD dissertation, Computer Science Department, Carnegie Mellon University, 1976.

5. L.D.Erman, F.Hayes-Roth, V.R.Lesser, D.R.Reddy, "The Hearsay-II Speech-Understanding System: Integrating Knowledge to Resolve Uncertainty," *ACM Computing Surveys*, Vol. 12, No. 2, June 1980, .

6. D.H.Klatt, K.N.Stevens, "On the Automatic Recognition of Continuous Speech: Implications from a Spectrogram Reading Experiment," *IEEE Transactions Audio Electroacoustics*, Vol. AU-21, 1973, pp. 210-217.

7. F.Alleva, "Unpublished Memos".

8. S.B.Davis, P.Mermelstein, "Comparison of Parametric Representations for Monosyllabic Word Recognition in Continuously Spoken Sentences," *IEEE Transactions on Acoustics, Speech, Signal Processing*, Vol. ASSP-28, No. 4, August 1980, pp. 357-366.

9. C.Myers, L.R.Rabiner, A.E.Rosenberg, "Performance Tradeoffs in Dynamic Time Warping Algorithms for Isolated Word Recognition," *TASSP*, Vol. ASSP-28, No. 6, December 1980, .

10. H.Sakoe, S.Chiba, "Dynamic Programming Optimization for Spoken Word Recognition," *IEEE Transactions on Acoustics, Speech, Signal Processing*, Vol. ASSP-26, No. 1, February 1978, pp. 43-49.

11. G.M.White, R.B.Neely, "Speech Recognition Experiments with Linear Prediction, Bandpass Filtering, and Dynamic Programming," *IEEE Transactions on Acoustics, Speech, Signal Processing*, Vol. ASSP-24, April 1976, pp. 183-188.

12. F.Itakura, "Minimum Prediction Residual Principle Applied to Speech Recognition," *IEEE Transactions on Acoustics, Speech, Signal Processing*, Vol. ASSP-23, No. 1, February 1975, pp. 67-72.

13. L.R.Rabiner,S.E.Levinson,A.E.Rosenberg,J.G.Wilpon, "Speaker-Independent Recognition of Isolated Words Using Clustering Techniques," *IEEE Transactions on Acoustics, Speech, Signal Processing*, Vol. ASSP-27, No. 4, August 1979, pp. 336-349.

14. L.R.Rabiner, A.E.Rosenberg, S.E.Levinson, "Considerations in Dynamic Time Warping Algorithms for Discrete Word Recognition," *IEEE Transactions on Acoustics, Speech, Signal Processing*, Vol. ASSP-26, No. 6, December 1978, pp. 575-582.

15. L.R.Rabiner, C.E.Schmidt, "Application of Dynmic Time Warping to Connected Digit Recognition," *IEEE Transactions on Acoustics, Speech, Signal Processing*, Vol. ASSP-28, No. 4, August 1980, pp. 377-388.

16. C.S.Myers, L.R.Rabiner, A.E.Rosenberg, "An Investigation of the Use of Dynamic Time Warping for Word Spotting and Connected Speech Recognition," *ICASSP 80 Proceedings Volume I*, IEEE ASSP, April 1980, pp. 173-177.

17. Y.Niimi, "Unpublished Progress Report", August 1980

18. S.K.Das, "Some Experiments in Discrete Utterance Recognition," *ICASSP 80 Proceedings Volume I*, IEEE ASSP, April 1980, pp. 178-181.

19. H.F.Silverman, N.R.Dixon, "State Constrained Dynamic Programming (SCDP) for Discrete Utterance Recognition," *ICASSP 80 Proceedings Volume I*, IEEE ASSP, April 1980, pp. 169-172.

20. L.R.Rabiner, "On Creating Reference Templates for Speaker Independent Recognition of Isolated Words," *IEEE Transactions on Acoustics, Speech, Signal Processing*, Vol. ASSP-26, No. 1, February 1978, pp. 34-42.

21. Z.Li, F.Alleva, R.Reddy, "Effect of Reference Set Selection on Speaker Dependent Speech Recognition," *J.Acoust.Soc.Am.*, Vol. 69, Suppl.1, May 1981, , soon to appear as CMU tech-report

22. E.L.Lehmann, *Nonparametrics: Statistical Methods Based on Ranks*, Holden-Day, Inc., 1975.

23. J.S.Briddle, "An Efficient Elastic Method for Detecting Given Words in Running Speech," *Proceedings of the British Acoustical Society Meeting*, British Acoustical Society, April 1973, pp. , paper 73SHC3